02 208
ARC

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００２年１２月１９日

出願番号
Application Number: 特願２００２－３６８２７６

[ST.10/C]: [ＪＰ２００２－３６８２７６]

出願人
Applicant(s): インターナショナル・ビジネス・マシーンズ・コーポレーション

２００３年　５月２０日

特許庁長官
Commissioner,
Japan Patent Office 太田信一郎

出証番号　出証特２００３－３０３７０１４

【書類名】　特許願

【整理番号】　JP9020208

【特記事項】　特許法第３６条の２第１項の規定による特許出願

【提出日】　平成14年12月19日

【あて先】　特許庁長官殿

【国際特許分類】　G06F 19/00

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　マイケル・エドワード・フール

【特許出願人】

【識別番号】　390009531

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】　100086243

【弁理士】

【氏名又は名称】　坂口　博

【代理人】

【識別番号】　100091568

【弁理士】

【氏名又は名称】　市位　嘉宏

【代理人】

【識別番号】　100108501

【弁理士】

【氏名又は名称】　上野　剛史

【復代理人】

【識別番号】　100110607

【弁理士】

【氏名又は名称】　間山　進也

【手数料の表示】

【予納台帳番号】　062651

【納付金額】　35,000円

【提出物件の目録】

【物件名】　外国語明細書　1

【物件名】　外国語図面　1

【物件名】　外国語要約書　1

【包括委任状番号】　9706050

【包括委任状番号】　9704733

【包括委任状番号】　0207860

【プルーフの要否】　要

【書類名】　外国語明細書

[Document name] Specification

[Title of the Invention]

A computer system for generating a data structure for information retrieval, a method thereof, a computer executable program for generating a data structure for information retrieval, a computer readable medium storing the program for generating a data structure for information retrieval, an information retrieval system, and a graphical user interface system

[Claims]

Claim 1 A computer system for generating data structures for information retrieval of documents stored in a database, said documents being stored as document-keyword vectors generated from a predetermined keyword list, and said document-keyword vectors forming nodes of a hierarchical structure imposed upon said documents, said computer system comprising:

a neighborhood patch generation part for generating groups of nodes having similarities as determined using a search structure, said neighborhood patch generation part including a part for generating a hierarchical structure upon said document-keyword vectors and a patch defining part for creating patch relationships among said nodes with respect to a metric distance between nodes; and

a cluster estimation part for generating cluster data of said document-keyword vectors using said similarities of patches.

Claim 2 The computer system of claim 1, wherein said computer system comprises a confidence determination part for computing inter-patch confidence values between said patches and intra-patch confidence values, and said cluster estimation part selects said patches depending on said inter-patch confidence values to represent clusters of said document-keyword vectors.

Claim 3 The computer system of claim 1, wherein said cluster estimation part estimates sizes of said clusters depending on said intra-patch confidence values.

Claim 4 A method for generating data structures for information retrieval of documents stored in a database, said documents being stored as document-keyword vectors generated from a predetermined keyword list, and said document-keyword vectors forming nodes of a hierarchical structure imposed upon said documents, said method comprising the steps of:
generating a hierarchical structure upon said document-keyword vectors and storing hierarchy data in an adequate storage area;
generating neighborhood patches of nodes having similarities as determined using levels of the hierarchical structure, and storing said patches in an adequate storage area;
invoking said hierarchy data and said patches to compute inter-patch confidence values between said patches and intra-patch confidence values, and storing said values as corresponding lists in an adequate storage area; and
selecting said patches depending on said inter-patch confidence values and said intra-patch confidence values to represent clusters of said document-keyword vectors.

Claim 5 The method according to claim 4 further comprising the step of estimating sizes of said clusters depending on said intra-patch confidence values.

Claim 6 A program for making a computer system execute a method for generating data structures for information retrieval of documents stored in

a database, said documents being stored as document-keyword vectors generated from a predetermined keyword list, and said document-keyword vectors forming nodes of a hierarchical structure introduced into said documents, said program making said computer system execute the steps of:

generating a hierarchical structure upon said document-keyword vectors and storing hierarchy data in an adequate storage area;

generating neighborhood patches consisting of nodes having similarities as determined using levels of the hierarchical structure, and storing said patches in an adequate storage area;

invoking said hierarchy data and said patches to compute inter-patch confidence values between said patches and intra-patch confidence values, and storing said values as corresponding lists in an adequate storage area; and

selecting said patches depending on said inter-patch confidence values and said intra-patch confidence values to represent clusters of said document-keyword vectors.

Claim 7 The method according to claim 6, further comprising the step of estimating sizes of said clusters depending on said intra-patch confidence values.

Claim 8 A computer readable medium storing a program for making a computer system execute a method for generating data structures for information retrieval of documents stored in a database, said documents being stored as document-keyword vectors generated from a predetermined keyword list, and said document-keyword vectors forming nodes of a hierarchical structure imposed upon said documents, said program making said computer system execute the steps of:

generating a hierarchical structure upon said document-keyword vectors a

nd storing hierarchy data in an adequate storage area;
generating neighborhood patches consisting of nodes having similarities as determined using levels of the hierarchical structure, and storing sa id patch list in an adequate storage area;
invoking said hierarchy data and said patches to compute inter-patch con fidence values between said patches and intra-patch confidence values, a nd storing said values as corresponding lists in an adequate storage are a; and
selecting said patches depending on said inter-patch confidence values a nd said intra-patch confidence values to represent clusters of said docu ment-keyword vectors.

Claim 9 The method according to claim 8, further comprising the step of estimating sizes of said clusters depending on said intra-patch confiden ce values.

Claim 10 An information retrieval system for of documents stored in a da tabase, said documents being stored as document-keyword vectors generate d from a predetermined keyword list, and said document-keyword vectors f orming nodes of a hierarchical structure imposed upon said documents, sa id system comprising:
a neighborhood patch generation part for generating groups of nodes havi ng similarities as determined using a hierarchical structure, said patch generation part including a part for generating a hierarchical structur e upon said document-keyword vectors and a patch defining part for creat ing patch relationships among said nodes with respect to a metric distan ce between nodes; and
a cluster estimation part for generating cluster data of said document-k eyword vectors using said similarities of patches; and

a graphical user interface part for presenting said estimated cluster da ta on a display means.

Claim 11 The computer system of claim 10, wherein said information retri eval system comprises a confidence determination part for computing inte r-patch confidence values between said patches and intra-patch confidenc e values, and said cluster estimation part selects said patches dependin g on said inter-patch confidence values to represent clusters of said do cument-keyword vectors.

Claim 12 The system of claim 10, wherein said cluster estimation part es timates sizes of said clusters depending on said intra-patch confidence values.

Claim 13 The system of claim 10, wherein said system further comprises a user query receiving part for receiving said query and extracting data for information retrieval to generate a query vector, and an information retrieval part for computing similarities between said document-keyword vectors and said query vector to select said document-keyword vectors.

Claim 14 The system of claim 10, wherein said clusters are estimated usi ng said retrieved document-keyword vectors with respect to said user inp ut query.

Claim 15 A graphical user interface system for graphically presenting es timated clusters on a display device in response to a user input query, said graphical user interface system comprising:
a database for storing documents;
a computer for generating document-keyword vectors for said documents st

ored in said database and for estimating clusters of documents in respon se to said user input query; and

a display for displaying on screen said estimated clusters together with confidence relations between said clusters and hierarchical information pertaining to cluster size.

Claim 16 The graphical user interface system of claim 15, wherein said c omputer comprises:

a neighborhood patch generation part for generating groups of nodes havi ng similarities as determined using a search structure, said neighborhoo d patch generation part including a part for generating a hierarchical s tructure upon said document-keyword vectors and a patch defining part fo r creating patch relationships among said nodes with respect to a metric distance between nodes; and

a cluster estimation part for generating cluster data of said document-k eyword vectors using said similarities of patches.

Claim 17 The graphical user interface system of claim 15, wherein said c omputer comprises a confidence determination part for computing inter-pa tch confidence values between said patches and intra-patch confidence va lues, and said cluster estimation part selects said patches depending on said inter-patch confidence values to represent clusters of said docume nt-keyword vectors and said cluster estimation part estimates sizes of s aid clusters depending on said intra-patch confidence values.

[Detailed Explanation of Invention]

[Field of Invention]

The present invention relates to information retrieval from a large data base, and more particularly relates to a computer system for generating

a data structure for information retrieval, a method thereof, a computer executable program for generating a data structure for information retrieval, a computer readable medium storing the program for generating a data structure for information retrieval, an information retrieval system, and a graphical user interface system.

[Background of the Art]

Recently, information processing systems are increasingly expected to handle large amounts of data such as, for example, news data, client information, patent information, and stock market data. Users of such databases find it increasingly difficult to search for desired information quickly and effectively with sufficient accuracy. Therefore, timely, accurate, and inexpensive detection of documents from large databases may provide very valuable information for many types of businesses. In addition, sometimes users wish to obtain further information related to data retrieved, such as cluster information in the database, and the interrelationships among such clusters.

Typical methods for detecting clusters rely upon a measure of similarity between data elements; such methods based on similarity search have been proposed so far as summarized below.

Similarity search (also known as proximity search) is one in which items of a database are sought according to how well they match a given query element. Similarity (or rather, dissimilarity) is typically modeled using some real- or integer-valued distance 'metric' dist: that is,

(1) dist(p, q) ≧ 0 for all p, q (non-negativity);

(2) dist(p, q) = dist(q, p) for all p, q (symmetry);

(3) dist(p, q) = 0 if and only if p = q;

(4) dist(p, q) + dist(q, r) ≧ dist(p, r) for all p, q, r (triangle inequality).

Any set of objects for which such a distance function exists is called a metric space. A data structure that allows a reduction in the number of distance evaluations at query time is known as an index. Many methods for similarity queries have been proposed. Similarity queries on metric spaces are of two general types, as stated below:
(A) k-nearest-neighbor query: given a query element q and a positive integer k, report the k closest database elements to q.
(B) range query: given a query element q and a distance r, report every database item p such that dist(p, q) ≦ r.

For large databases, it is too expensive to perform similarity queries by means of explicitly computing the distances from the query element to every database element. Previous computation and storage of all distances among database elements is also too expensive, as this would require time and space proportional to the square of the number of database elements (that is, quadratic time and space). A more practical goal is to construct a search structure that can handle queries in sub-linear time using sub-quadratic storage and preprocessing time.

A. Review of Vector Space Models

Current information retrieval methods often uses vector space modeling to represent the documents of databases. In such vector space models, each document in the database under consideration is associated with a vector, each coordinate of which represents a keyword or attribute of the document; details of the vector space models are provided elsewhere (Gerald Salton, The SMART Retrieval System - Experiments in Automatic Document

Processing, Prentice-Hall, Englewood Cliffs, NJ, USA, 1971).

B. Brief Survey of Similarity Search Structures

A great variety of structures have been proposed over the past thirty ye ars for handling similarity queries. The majority of these are spatial i ndices, which require that the object set be modeled as a vector of d re al-valued attributes. Others are 'metric' indices, which make no assump tions on the nature of the database elements other than the existence of a distance metric, and are therefore more widely-applicable than spatia l search structures. For recent surveys of search structures for multi-d imensional vector spaces and metric spaces, see Gaede et al. (Volker Gae de and Oliver Gunther, Multidimensional Access Methods, ACM Computing Su rveys, 30, 2, 1998, pp. 170-231.), and Chavez et al. (Edgar Chavez, Gonz alo Navarro, Ricardo Baeza-Yates and Jose L. Marroquin, Searching in met ric spaces, ACM Computing Surveys 33, 3, 2001, pp. 273-321.).

The practicality of similarity search, whether it be on metric data or v ector data, is limited by an effect often referred to as the 'curse of dimensionality'. Recent evidence suggests that for the general problem o f computing nearest-neighbor or range queries on high-dimensional data s ets, exact techniques are unlikely to improve substantially over a seque ntial search of the entire database, unless the underlying distribution of the data set has special properties, such as a low fractal dimension, low intrinsic dimension, or other properties of the distribution.

For more information regarding data dimension and the curse of dimension ality, see (for example) Chavez et al. (op cito)), Pagel et al. (Bernd-U we Pagel, Flip Korn and Christos Faloutsos, Deflating the dimensionality curse using multiple fractal dimensions, Proc. 16th International Confe

rence on Data Engineering (ICDE 2000), San Diego, USA, IEEE CS Press, 2000, pp. 589-598.), Pestov (Vladimir Pestov, On the geometry of similarity search: dimensionality curse and concentration of measure, Information Processing Letters, 73, 2000, pp. 47-51.), and Weber et al. (Roger Weber, Hans-J. Schek and Stephen Blott, A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, Proc. 24th VLDB Conference, New York, USA, 1998, pp. 194-205).

C. Brief Survey of Approximate Similarity Searching

In an attempt to circumvent the curse of dimensionality, researchers have considered sacrificing some of the accuracy of similarity queries in the hope of obtaining a speed-up in computation. Details of these techniques are provided elsewhere, for example, by Indyk et al. (P. Indyk and R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, Proc. 30th ACM Symposium on Theory of Computing, Dallas, 1998, pp. 604-613.), and Ferhatosmanoglu et al. (Hakan Ferhatosmanoglu, Ertem Tuncel, Divyakant Agrawal and Amr El Abbadi, Approximate nearest neighbor searching in multimedia databases, Proc. 17th International Conference on Data Engineering (ICDE), Heidelberg, Germany, IEEE CS Press, 2001, pp. 503-514.); for metric spaces, by Ciaccia et al. (Paolo Ciaccia and Marco Patella, PAC nearest neighbor queries: approximate and controlled search in high-dimensional and metric spaces, Proc. 16th International Conference on Data Engineering (ICDE 2000), San Diego, USA, 2000, pp. 244-255; Paolo Ciaccia, Marco Patella and Pavel Zezula, M-tree: an efficient access method for similarity search in metric spaces, Proc. 23rd VLDB Conference, Athens, Greece, 1997, pp. 426-435.) and Zezula et al. (Pavel Zezula, Pasquale Savino, Giuseppe Amato and Fausto Rabitti, Approximate similarity retrieval with M-trees, The VLDB Journal, 7, 1998, pp. 275-293.). However, these methods all suffer from deficiencies that limi

t their usefulness in practice. Some make unrealistic assumptions concer ning the distribution of the data; others cannot effectively manage the trade-off between accuracy and speed.

D. Spatial Approximation Sample Hierarchy (SASH)

An approximate similarity search structure for large multi-dimensional d ata sets that allows significantly better control over the accuracy-spee d tradeoff is the spatial approximation sample hierarchy (SASH), describ ed in Houle (Michael E. Houle, SASH: a spatial approximation sample hier archy for similarity search, IBM Tokyo Research Laboratory Research Repo rt RT-0446, 18 pages, February 18, 2002) and Houle, Kobayashi and Aono (Japanese Patent Application No. 2002-037842). The SASH requires a simila rity function satisfying the conditions of a distance metric, but otherw ise makes no assumptions regarding the nature of the data. Each data ele ment is given a unique location within the structure, and each connectio n between two elements indicates that they are closely related. Each lev el of the hierarchy consists of a random sample of the elements, the sam ple size at each level roughly double that of the level immediately abov e it. The structure is organized in such a way that the elements located closest to a given element v are those that are most similar to v. In p articular, the node corresponding to v is connected to a set of its near neighbors from the level above, and also to a set of items from the lev el below that choose v as a near neighbor.

E. Review of Clustering Techniques

The term clustering refers to any grouping of unlabeled data according t o similarity criteria. Traditional clustering methods can generally be c lassified as being either partitional or hierarchical. Hierarchical tech niques produce a tree structure indicating inclusion relationships among

groups of data (clusters), with the root of the tree corresponding to t he entire data set. Partitional techniques typically rely on the global minimization of classification error in distributing data points among a fixed number of disjoint clusters. In their recent survey, Jain, Murty and Flynn (A. K. Jain, M. N. Murty and P. J. Flynn, Data clustering: a r eview, ACM Computing Surveys 31, 3, 1999, pp. 264-323.) argue that parti tional clustering schemes tend to be less expensive than hierarchical on es, but are also considerably less flexible. Despite being simple, fast (linear observed time complexity), and easy to implement, even the well-known partitional algorithm K-means and its variants generally do not pe rform well on large data sets. Partitional algorithms favor the generati on of isotropic (rounded) clusters, but are not well-suited for finding irregularly-shaped ones.

F. Hierarchical agglomerative clustering

In a hierarchical agglomerative clustering, each data point is initially considered to constitute a separate cluster. Pairs of clusters are then successively merged until all data points lie in a single cluster. The larger cluster produced at each step contains the elements of both merge d subclusters; it is this inclusion relationship that gives rise to the cluster hierarchy. The choice of which pairs to merge is made so as to m imimize some inter-cluster distance criterion.

G. Shared-neighbor methods

One of the criticisms of simple distance-based agglomerative clustering methods is that they are biased towards forming clusters in regions of h igher density. Well-associated groups of data in regions of low density risk not being discovered at all, if too many pairwise distances fall be low the merge threshold. More sophisticated (and expensive) distance mea

sures for agglomerative clustering have been proposed, that take into account the neighborhoods of the data elements. Jarvis et al. (R. A. Jarvis and E. A. Patrick, Clustering using a similarity measure based on shared nearest neighbors, IEEE Transactions on Computers C-22, 11, Nov. 1973, pp. 1025-1034.) defined a merge criterion in terms of an arbitrary similarity measure dist and fixed integer parameters k > r > 0, in which two data elements find themselves in the same cluster if they share at least a certain number of nearest neighbors. The decision as to whether to merge clusters thus does not depend on the local density of the data set, but rather as to whether there exists a pair of elements, one drawn from each, that share a neighborhood in a substantial way.

Jarvis and Patrick' s method (op. cito) is agglomerative, and resembles the single-link method in that it tends to produce irregular clusters via chains of association. More recent variants have been proposed in an attempt to vary the qualities of the clusters produced: for example, by Guha et al. (S. Guha, R. Rastogi and K. Shim, ROCK: a robust clustering algorithm for categorical attributes, Information Systems 25, 5, 2000, pp. 345-366.); by Ertoz et al. (Levent Ertoz, Michael Steinbach and Vipin Kumar, Finding topics in collections of documents: a shared nearest neighbor approach, University of Minnesota Army HPC Research Center Preprint 2001-040, 8 pages, 2001.); by Ertoz et al. (Levent Ertoz, Michael Steinbach and Vipin Kumar, A new shared nearest neighbor clustering algorithm and its applications, Proc. Workshop on Clustering High Dimensional Data and its Applications (in conjunction with 2nd SIAM International Conference on Data Mining), Arlington, VA, USA, 2002, pp. 105-115.); by Daylight Chemical Information Systems Inc., in URL address (http://www.daylight.com/); and by Barnard Chemical Information Ltd., in URL address (http://www.bci.gb.com/). Nonetheless, all variants still exhibit the main ch

aracteristics of agglomerative algorithms, in that they allow the format ion of large irregularly-shaped clusters with chains of association brid ging poorly-associated elements.

H. Review of Methods for Dimension Reduction

Latent semantic indexing (LSI) is a vector space model-based algorithm f or reducing the dimension of the document ranking problem; see Deerweste r et al. (Scott Deerwester, Susan T. Dumais, George W. Furnas, Richard H arshman, Thomas K. Landauer, Karen E. Lochbaum, Lynn A. Streeter, Comput er information retrieval using latent semantic analysis, U.S. Patent No. 4839853, filed Sept. 15, 1988, issued June 13, 1989; Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman, Indexing by latent semantic analysis, Journal of the American Society f or Information Science, 41, 6, 1990, pp. 391-407.). LSI reduces the retr ieval and ranking problem to one of significantly lower dimension so tha t retrieval from very large databases can be performed more efficiently. Another dimension-reduction strategy due to Kobayashi et al. (Mei Kobay ashi, Loic Malassis, Hikaru Samukawa, Retrieval and ranking of documents from a database, IBM Japan, docket No. JP9-2000-0075, filed June 12, 20 00; Loic Malassis, Mei Kobayashi, Statistical methods for search engines , IBM Tokyo Research Laboratory Research Report RT-413, 33 pages, May 2, 2001.) provides a dimensional reduction method called COV, which uses t he covariance matrix of the document vectors to determine an appropriate reduced-dimensional space into which to project the document vectors. L SI and COV are comparable methods for information retrieval; for some da tabases and some queries, LSI leads to slightly better results than COV, while for others, COV leads to slightly better results.

[Problem to be Solved by Invention]

Conventional cluster detection based on distances has other inconvenienc es as described herein below:

The usual clustering methods for machine learning contexts are designed to find major groupings within data sets. Here, a method is considered g ood if the clusters allow unknown points to be classified with high accu racy. However, in data mining contexts, the major clusters of the data a re often well understood by users, and it is the smaller, minor clusters that have the potential of revealing valuable nuggets of information. E xisting clustering techniques based on partition or agglomeration are la rgely ineffective at separating out small data clusters from their backg round.

There is another inconvenience that massive text databases are typically partitioned into smaller collections in order to increase the efficienc y of information retrieval operations. This distribution is usually perf ormed so that the largest clusters in the data set remain intact within a single database. However, partition methods that focus on major cluste rs may cause valuable minor clusters to be dispersed among several datab ases. Identifying minor clusters as well as major clusters can lead to p artitions that more effectively preserve minor clusters.

As described before, some users of clustering tools are often interested in knowing the relationships among the clusters produced by the tool. H ierarchical clustering algorithms attempt to fill this need by producing a nested collection of clusters, with a single cluster containing the e ntire data set at the top, and the smallest clusters at the bottom. Howe ver, many of these clusters may exist only as a byproduct of the hierarc hical organization, and have no useful interpretation of their own. User

s would primarily expect each cluster reported by a data mining tool to have some independent conceptual interpretation. Once a set of meaningful clusters has been identified, users would likely be interested in knowing of any overlap or inclusion relationships among them.

In addition, in multi-dimensional settings it is very difficult to represent or describe the associative qualities of data clusters in a way that is easy for users to understand. When browsing clustered data, users need to be able to assess the degree of cohesion and prominence of clusters at a glance.

With respect to hardware resources for the retrieval, clustering has generally been viewed as desirable yet impractical for data mining applications, due to the computation cost associated with achieving high-quality clusters when the data sets are very large. There is a tremendous demand for tools that can provide some insight into the organization of large data sets in reasonable time on an ordinary computer.

As described above, many methods has been proposed so far. Nevertheless, a novel data structure suitable for information retrieval with high efficiency, high speed together with sufficient scalability has been required in the art.

[Means to Solve Problem]

The present invention hereby proposes a system and a method for information retrieval and data mining of large text databases, based on the identification of clusters of elements (e.g. documents) that exhibit a high degree of mutual similarity relative to their background.

In the present invention, profiles of clusters can be graphically displayed to the user, providing immediate visual feedback as to their quality and significance. Cluster attributes such as size and quality are assessed automatically by the system. The system also allows users to query the data set for clusters without the need for a precomputed global clustering. Scalability is achieved by means of dimensional reduction techniques, random sampling, and the use of data structures supporting approximate similarity search.

The present invention provides the above-described novel information retrieval features by improving detection efficiency of minor clusters while preserving such minor clusters. The novel information retrieval according to the present invention allows the interrelations of the clusters to be expressed as a graph structure to aid user understanding of the clusters. The present invention further makes it possible to improve the computation scalability of the computation of information retrieval.

The above aspects are provided by a system and methods for information retrieval and data mining of text databases, using shared neighbor information to determine query clusters. The clustering method assesses the level of mutual association between a query element (which may or may not be an element of the data set) and its neighborhood within the data set. The association between two elements is considered strong when the elements have a large proportion of their nearest neighbors in common. In contrast with previous methods making use of shared-neighbor information, the proposed methods are based on the new and original concepts of inter-cluster association confidence (CONF) and intra-cluster association self-confidence (SCONF).

According to the present invention, a computer system is provided for generating data structures for information retrieval of documents stored in a database, the documents being stored as document-keyword vectors generated from a predetermined keyword list, and the document-keyword vectors forming nodes of a hierarchical structure imposed upon the documents.

The computer system comprises:

a neighborhood patch generation part for generating groups of nodes having similarities as determined using a search structure, the patch generation part including a part for generating a hierarchical structure upon the document-keyword vectors and a patch defining part for creating patch relationships among said nodes with respect to a metric distance between nodes; and

a cluster estimation part for generating cluster data of the document-keyword vectors using the similarities of patches.

According to the present invention, the computer system comprises a confidence determination part for computing inter-patch confidence values between the patches and intra-patch confidence values, and the cluster estimation part selects the patches depending on the inter-patch confidence values to represent clusters of the document-keyword vectors.

According to the present invention, the cluster estimation part estimates sizes of the clusters depending on the intra-patch confidence values.

According to the present invention, a method is provided for generating data structures for information retrieval of documents stored in a database, the documents being stored as document-keyword vectors generated from a predetermined keyword list, and the document-keyword vectors forming nodes of a hierarchical structure imposed upon the documents. The meth

od comprises the steps of:

generating a hierarchical structure upon the document-keyword vectors, a nd storing hierarchy data in an adequate storage area;

generating neighborhood patches consisting of nodes having similarities as determined using levels of the hierarchical structure, and storing th e patches in an adequate storage area;

invoking the hierarchy data and the patches to compute inter-patch confi dence values between the patches and intra-patch confidence values, and storing the values as corresponding lists in an adequate storage area; a nd

selecting the patches depending on the inter-patch confidence values and said intra-patch confidence values to represent clusters of the documen t-keyword vectors.

According to the present invention, a program may be provided for making a computer system execute a method for generating data structures for i nformation retrieval of documents stored in a database, the documents be ing stored as document-keyword vectors generated from a predetermined ke yword list, and the document-keyword vectors forming nodes of a hierarch ical structure imposed upon the documents. The program makes the compute r system execute the steps of:

generating a hierarchical structure upon the document-keyword vectors an d storing hierarchy data in an adequate storage area;

generating neighborhood patches consisting of nodes having similarities as determined using levels of the hierarchical structure, and storing th e patches in an adequate storage area;

invoking the hierarchy data and the patches to compute inter-patch confi dence values between the patches and intra-patch confidence values, and storing the values as corresponding lists in an adequate storage part; a

nd

selecting the patches depending on the inter-patch confidence values and intra-patch confidence values to represent clusters of the document-keyword vectors.

According to the present invention, a computer readable medium may be provided for storing a program for making a computer system execute a method for generating data structures for information retrieval of documents stored in a database, the documents being stored as document-keyword vectors generated from a predetermined keyword list, and the document-keyword vectors forming nodes of a hierarchical structure imposed upon the documents. The program makes the computer system execute the steps of:

generating a hierarchical structure upon the document-keyword vectors and storing hierarchy data in an adequate storage area;

generating neighborhood patches consisting of nodes having similarities as determined using levels of the hierarchical structure, and storing the patch list in an adequate storage area;

invoking the hierarchy data and the patches to compute inter-patch confidence values between the patches and intra-patch confidence values, and storing the values as corresponding lists in an adequate storage area; and

selecting the patches depending on the inter-patch confidence values and intra-patch confidence values to represent clusters of the document-keyword vectors.

According to the present invention an information retrieval system may be provided for documents stored in a database, the documents being stored as document-keyword vectors generated from a predetermined keyword list, and the document-keyword vectors forming nodes of a hierarchical stru

cture imposed upon the documents. The system comprises:
a neighborhood patch generation part for generating groups of nodes having similarities as determined using a hierarchical structure, the patch generation part including a part for generating a hierarchical structure upon the document-keyword vectors and a patch defining part for creating patch relationships among said nodes with respect to a metric distance between nodes; and
a cluster estimation part for generating cluster data of the document-keyword vectors using the similarities of patches; and
a graphical user interface part for presenting the estimated cluster data on a display means.

According to the present invention, the information retrieval system comprises a confidence determination part for computing inter-patch confidence values between the patches and intra-patch confidence values, and the cluster estimation part selects the patches depending on the inter-patch confidence values to represent clusters of the document-keyword vectors. According to the present invention, the cluster estimation part estimates sizes of the clusters depending on the intra-patch confidence values. According to the present invention, the system further comprises a user query receiving part for receiving the query and extracting data for information retrieval to generate a query vector, and an information retrieval part for computing similarities between document-keyword vectors and the query vector to select the document-keyword vectors. The clusters are estimated using the retrieved document-keyword vectors with respect to the user input query.

According to the present invention, a graphical user interface system for graphically presenting estimated clusters on a display device in respo

nse to a user input query may be provided. The graphical user interface system comprising:

a database for storing documents;

a computer for generating document-keyword vectors for the documents stored in the database and for estimating clusters of documents in response to the user input query; and

a display for displaying on screen the estimated clusters together with confidence relations between the clusters and hierarchical information pertaining to cluster size.

According to the present graphical user interface, the computer comprises a neighborhood patch generation part for generating groups of nodes having similarities as determined using a search structure, the neighborhood patch generation part including a part for generating a hierarchical structure upon the document-keyword vectors and a patch defining part for creating patch relationships among the nodes with respect to a metric distance between nodes; and

a cluster estimation part for generating cluster data of the document-keyword vectors using the similarities of patches. Further according to the present invention, the computer comprises a confidence determination part for computing inter-patch confidence values between the patches and intra-patch confidence values, and the cluster estimation part selects the patches depending on the inter-patch confidence values to represent clusters of the document-keyword vectors and the cluster estimation part estimates sizes of the clusters depending on the intra-patch confidence values.

[Embodiment of the Invention]

Part I. Essential Processes of the Method

Hereinafter, the present invention will be explained in the context of i

nformation retrieval of documents; however, the present invention is not limited thereto and the algorithm of the present invention can be adapted for any application for which a pairwise dissimilarity measure is used that satisfies the properties of a distance metric (with the possible exception of the triangle inequality), and for which each data element has keywords or other information that can be used for annotation purposes. One example of such an application is a data mining system for multimedia databases (e.g., databases with contents which consist of text, audio, video, still images, graphics images, graphics videos, and/or gif animations, etc.) having contents for which such a pairwise dissimilarity metric exists.

A flowchart of the general method according to the present invention is shown in Fig. 1. Although the present invention is primarily explained using an application to for texts, a person skilled in the art may understand that the methods of the present invention are easily adapted to any database with contents which may be modeled with a clearly defined metric that enables computation of distances between any two elements so that pairs of elements which are "closer" (with respect to the metric) are more similar than pairs of elements that are "further apart".

The method of the present invention begins from the step S10 where documents in a database are transformed into vectors using the vector space model. Next, the method generates in the step S12 a SASH similarity search structure for the data stored in the database. Next, for every element of the database, the SASH structure is used in the step S14 to compute a neighborhood patch consisting of a list of those database elements most similar to it. These patches are then stored in an adequate memory area.

In the step S16, a list of self-confidence values, hereafter referred to as SCONF values, are computed for every stored patch. These SCONF values are used to compute relative self-confidence values, hereafter referred to as RSCONF values, that are in turn used to determine the size of the best subset of each patch (which is itself also a patch) to serve as a cluster candidate. Next, the method proceeds to the step S18, at which confidence values, hereafter referred to as CONF values, are used to eliminate redundant cluster candidates. The method then proceeds to the Step S20 for further selection of those cluster candidates having at least a desired minimum value of RSCONF as the final clusters, and storing these selected clusters in an adequate memory. The method further proceeds to the step S22 to display to the user by a GUI interface on a computer screen a graph indicating the interrelationships among the clusters. The method of Fig. 1 further comprises sub-steps for performing each step of Fig. 1, and the sub-steps will hereinafter be described in detail.

<Computation of Document-Keyword Vectors>

Document-keyword vectors may be computed from given keywords and documents using any of several known techniques. In a particular embodiment of the present invention, appropriate weighting is used to digitize the documents; details of the digitization has been provided elsewhere (e.g. Salton et al., op. cito), and therefore are not explained in the present invention.

<SASH Construction and Usage>

Fig. 2 shows a general procedure for constructing the hierarchical structure of the document-keyword vectors known as a spatial approximation sample hierarchy, or SASH. The process begins at the step S28 after receiv

ing the result of the step S10 of Fig. 1 to generate a random assignment of vectors to nodes of the SASH using, for example, any well-known random number generating program. The levels are numbered from 0 to h, where each level contains roughly twice as many vector nodes as the one following it. The level numbered 0 contains roughly half the vector nodes of the data set, and the level numbered h contains a single node, called the top node. The top node of the SASH structure is determined randomly using any random number generation means included elsewhere in the computer system. Next, in the step S30, a hierarchy level reference L is initialized to h. The process proceeds to the step S32 to decrease the hierarchy level L by 1 and in the step S34 level L nodes are connected to a set of level L+1 nodes depending on distances between the nodes. In the above connection, the nodes at level L+1 become parent nodes and the nodes at level L become child nodes. The connection is performed by choosing parents of a node from level L from among the closest nodes from level L+1, and then connecting these parent-child node pairs so that each parent is connected to a predetermined number of its closest children. Further details on how the connections are performed are given elsewhere, by Houle et al. (op. cito). The process proceeds to the step S36 and determines whether or not the hierarchy level reaches to the lowest level (0), and if so (yes), the construction of the SASH is completed and the SASH structure is stored in an adequate memory area such as memory or a hard disk. The process continues to the step S38 to construct patches of nodes. If not so (no), the process reverts to the step S32 to repeat until an affirmative result in the step S36 is obtained.

In the step 38, the stored SASH structure is used according to the present invention to generate a patch for every element of the database. A patch for a given element q with respect to a subset R of the database is

a set of neighboring elements of q drawn from R, according to a predeter mined measure of similarity dist. In the described embodiment for constr ucting the SASH, each node in the database are labeled with its hierarch y level, and the patch for each node is of a predetermined, fixed size, and is computed with respect to the set of all nodes at the same level o r greater. The present invention is not limited to constructing and stor ing only one patch per node; additional patches with respect to other no de sets may also be constructed and stored.

Fig. 3 shows an illustrative example of construction of the SASH structu re together with the structure of the patch created according to the pre sent invention. As described in Fig. 3, the vector nodes referred to by a patch can essentially belong to any of the SASH hierarchy levels at or above the level of the vector node upon which it is based. In addition, from among the nodes at these hierarchy levels, patches contain the nod es closest to the base node according to a predetermined "metric distan ce". The base node may be selected from any or all nodes included in th e hierarchical structure so as to provide global constructions of cluste rs; in an alternative embodiment of the present invention, the base node may be determined using a user inputted query so as to provide cluster information specifically about the queried base node, i.e., a retrieved document. The base node is represented by the star in Fig. 3 and the nod es in the patch are aligned with respect to the user query as shown in F ig. 3. The patch structure is also stored in an adequate memory area in the system described in detail hereinafter. In the present invention, th ese patches are further related in terms of confidence, as described bel ow.

<Computation of Confidences>

The method of the present invention uses a novel model for clustering th at borrows from both information retrieval and association rule discover y herein named the "patch model". The patch model assumes that data cl usters can be represented as the results of neighborhood queries based o n elements from the data set, according to some measure of (dis)similari ty appropriate to the domain. More formally, let S be a database of elem ents drawn from some domain D, and now, let "dist" be a pairwise dista nce function defined on D satisfying the properties of a metric, as defi ned earlier. Further now, let R be a subset of S. For any given query pa ttern $q \in D$, let NN(R, q, k) which denote a k-nearest neighbor set of q , drawn from R according to dist, and chosen subject to the following co nditions:

If $q \in R$, then NN(R, q, 1) = {q}, that is, if q is a member of the data set, then q is considered to be its own nearest neighbor.
$NN(R, q, k-1) \subset NN(R, q, k)$ for all $1 < k \leqq |R|$, that is, smaller neig hborhoods of q are strictly contained in larger neighborhoods.

These conditions take into account the possibility that q may have more than one distinct k-nearest neighbor set in R. The uniquely-determined s et NN(R, q, k) is referred as the k-patch of q (relative to R), or simpl y as one of the patches of q.

Fig. 4 illustrates a collection of patches (a 7-patch, a 12-patch, and a n 18-patch) of a database. The dashed circle represents the entire docum ent space.

Consider now the situation in which two potential clusters within R are represented by the two patches $C_i = NN(R, q_i, k_i)$ and $C_j = NN(R, q_j, k_j)$

. The relevance of C_j to C_i is assessed according to a natural confidence measure resembling that of association rule discovery proposed by Agrawal and Srikant (op. cito):

$$CONF(C_i,C_j) = |C_i \cap C_j|/|C_i| = |NN(R, q_i, k_i) \cap NN(R, q_j, k_j)|/k_i.$$

That is, the confidence is expressed as the proportion of elements forming C_i that also contribute to the formation of C_j. If the confidence value is small, the candidate C_j has little or no impact upon C_i; on the other hand, if the proportion is large, C_j is strongly related to C_i, possibly even subsuming it.

Fig. 5 shows an essential function of CONF to the clusters A and B which include 8 and 10 vectors, respectively. Two vectors are in the common intersection of A and B, and therefore when the function CONF is applied to the patches in the order A, B, that is, CONF (A, B), the result is 0.25 or 25%. When the function is applied in the order B, A, that is, CONF (B, A), the result is 0.2 or 20%. The function CONF can be applied to any two patches drawn from a common underling sample of the database.

The confidence measure can also be regarded as an example of a shared-neighbor distance metric. However, the uses to which the shared-neighbor information are put in this invention are very different from those of agglomerative clustering methods: whereas agglomerative methods use such metrics to decide whether two patches should be merged, the proposed method uses it to assess the quality of the level of association between two query patches.

<Computation of Intra-Cluster Association>

A natural assessment of association within patches is also possible in terms of the notion of confidence. Let C_q = NN(R, q, k) be a patch cluster candidate. Here the constituent patches of C_q is defined to be the set of those patches of the form C_v = NN(R, v, k), for all elements $v \in C_q$. If C_q has a high degree of internal association, then one can reasonably expect strong relationships between C_q and its constituent patches. On the other hand, low internal association would manifest itself as weak relationships between C_q and its constituent patches. Therefore, internal association within a patch cluster candidate in terms of its self-confidence is obtained and is defined as the average confidence of the candidate patches with respect to its constituent patches:

$$SCONF(C_q) = (1 / |C_q|) * \Sigma_{v \in Cq,\ |Cv|=|Cq|}\ CONF(C_q, C_v)$$
$$= (1 / k^2) * \Sigma_{v \in Cq}\ |NN(R, q, k) \cap NN(R, v, k)|.$$

A self-confidence value of 1 indicates perfect association among all elements of a cluster, whereas a value approaching 0 indicates little or no internal association.

<Cluster Boundary Determination Using Intra-cluster Confidence>
Herein assume for the moment that the subject node q is associated with some cluster within R that we want to estimate. Using the notion of self-confidence, the process determines the k-patch based at q that best describes this cluster, over some range of interest $a \leqq k \leqq b$. The ideal patch would be expected to consist primarily of cluster elements, and to have a relatively high self-confidence, whereas larger patches would be expected to contain many elements from outside the cluster and to have a relatively low self-confidence. The evaluation focuses on two patches: an inner patch $C_{q,k}$ = NN(R, q, k) of size k indicating a candidate patc

h cluster, and an outer patch $C_{q,\ \phi(k)}$ = NN(R, q, ϕ(k)) of size ϕ(k) > k that provides the local background against which the suitability of the inner patch will be judged.

For a given choice of k, the neighbor sets of each element of the outer patch are examined. Consider the neighbor pair (v,w) with v in the outer patch, and w a member of the outer constituent patch NN(R, v, ϕ(k)). If v also lies in the inner patch, and w is a member of the inner constituent patch NN(R, v, k), then herein (v,w) is referred to as an inner neighbor pair.

If w is a member of the outer patch, then the pair (v,w) contributes to the self-confidence of the outer patch, thereby undermining the choice of the inner patch as the descriptor for the cluster based at q. If w is also a member of the inner patch, and (v,w) is an inner pair, then the pair contributes to the self-confidence of the inner patch, thereby strengthening the association between v and q.

Essentially, the k-patch best describing the cluster containing q would achieve as below:
i) a high proportion of inner pairs that contribute to the self-confidence of the inner patch, and
ii) a high proportion of neighbor pairs (not necessarily inner) that do not contribute to the self-confidence of the outer patch.

A high proportion of the former kind indicates a high level of association within the k-patch, whereas a high proportion of the latter kind indicates a high level of differentiation with respect to the local background. As both considerations are equally important, these proportions shou

ld be accounted for separately. The above considerations has been taken into account by maximizing, over all choices of k in the range $a \leqq k \leqq b$, the sum of these two proportions: that is, $SCONF(C_{q,k})$ and $1 - SCONF(C_{q,\phi(k)})$.

The relative self-confidence maximization (RSCM) problem can thus be for mulated as follows:

$\max_{a \leqq k \leqq b} RSCONF(C_{q,k}, \phi)$,

where

$RSCONF(C_{q,k}, \phi) = SCONF(C_{q,k}) - SCONF(C_{q, \phi(k)})$

$= SCONF(NN(R, q, k)) - SCONF(NN(R, q, \phi(k)))$,

wherein RSCONF is referred to as the relative self-confidence of the k-p atch $C_{q,k}$ with respect to R and ϕ. The k-patch at which the maximum is attained shall be referred to as the query cluster of q over this range. RSCM can be viewed as a form of maximum likelihood estimation (MLE), in which neighbor pairs are classified as either supporting or not support ing the choice of the inner patch as the query cluster.

Fig. 6 shows a sample pseudo-code listing for computing SCONF included i n the method of the present invention as part of a patch profile of a qu ery element, assuming that the neighbor lists $NN(R, q, \phi(b))$ and $NN(R, v, \phi(b))$ are already available for all v c $NN(R, q, \phi(b))$. Instead of producing $SCONF(NN(R, q, k))$ via direct computation, it is obtained from $SCONF(NN(R, q, k-1))$ by computing the differential resulting from the e xpansion of the patch by one item.

In the present invention, the RSCM method as presented allows for many v ariations in the way the outer patch size depends on the value of k (k i s integer.). Although the simple choice $\phi(k) = 2k$ is ideal in that it p

rovides the best balance between membership and non-membership of outer patch elements with respect to the inner patch, other considerations may influence the choice of $\phi(k)$. For example, the cost of computing boundary sharpness values may encourage the use of a maximum patch size $m < 2b$. In this case, the outer patch size could be chosen to be $\phi(k)=\min\{2k, m\}$, provided that the smallest ratio m/b between outer and inner patch sizes is still substantially greater than 1.

In the present invention, the design of the RSCM method assumes that internal cluster association is equally important as external differentiation. However, in the present invention, different weightings can be given to the internal and external contributions to the relative self-confidence value; that is, one can instead maximize functions of the form

$$RSCONF'(C_{q,k}, \phi) = w_1\, SCONF(C_{q,k}) - w_2\, SCONF(C_{q,\phi(k)}),$$

for real-valued choices of weights $0 < w_1$ and $0 < w_2$.

In the present stage, each stored patch $C_{q,m} = NN(R, q, m)$ is associated with a list of self-confidence values $SCONF(C_{q,k})$ for each sub-patch $C_{q,k} = NN(R, q, k)$ of $C_{q,m}$, for all values of k in the range $1 \leqq k \leqq m$. The data constructions hereinafter referred to as the SCONF list, shown in Fig. 7, may be recorded in an adequate storage means such as a hard disk or a memory to be referred to by the cluster selection function of the present invention.

Further variation of the present invention is to save the cost of computation. The cost of computing RSCONF values grows quadratically as the maximum outer patch size increases. This cost restricts the size of cluste

rs that can be discovered in practice using the RSCM method directly on the full data set. However, these restrictions can be circumvented throu gh the use of random sampling techniques. Instead of accommodating large clusters by adjusting the limits of the range $a \leqq k \leqq b$ over which th e RSCM problem is solved, one can instead search for patches of sizes in a fixed range, taken relative to a collection of data samples of varyin g size.

To understanding the above variation, the relationship between a uniform random sample $R \subseteq S$ and a hypothetical query cluster NN(S, q, c), for some large value of c. The intersection of NN(S, q, c) and R produces a patch NN(R, q, k), where $k = |NN(S, q, c) \cap R|$. The patch NN(R, q, k) s erves as a proxy for NN(S, q, c) with respect to the sample R - the choi ce of NN(R, q, k) as a query cluster for q in R can be taken as an indic ation of the appropriateness of NN(S, q, c) as a query cluster for q wit h respect to the entire data set.

If $a \leqq k \leqq b$, then the proxy patch will be evaluated by the RSCM metho d. Otherwise, if k does not lie between a and b, the patch will not be e valuated. In terms of the unknown "true" cluster size c, bounds on the probability of the proxy patch not being evaluated can be derived using standard Chernoff bound techniques, as described (for example) in Motwa ni and Raghavan (R. Motwani and P. Raghavan, Randomized Algorithms, Camb ridge University Press, New York, USA, 1995.):

$$E[k] = \mu = c\,|R| \,/\, |S|$$

$$\Pr\,[k < a \mid c] \leqq e^{-\mu}\,[e\mu \,/\, (a-1)]^{a-1}$$

$$\Pr\,[k > b \mid c] \leqq e^{-\mu}\,[e\mu \,/\, (b+1)]^{b+1}.$$

One can use these bounds as a guide in choosing appropriate values of a and b, as well as a collection of samples of appropriate sizes, so that for a desired probability for sufficiently-large c, at least one proxy patch has size between a and b for at least one of the samples.

As an illustrative example, consider a collection of uniform random samples $\{R_0, R_1, R_2, \ldots\}$ such that $|R_i| = |S| / 2^i$ for $i \geqq 0$. Now, let $NN(R_i, q, k_i)$ be the proxy patch of $NN(S, q, c)$, where c is an unknown value guaranteed to be at least 25. If the limits a = 25 and b = 120 are chosen, then for at least one sample R_i, the expected size $\mu_i = E[k_i]$ of its proxy patch must lie in the range $44 \leqq \mu_i \leqq 88$. Applying the bounds stated above, when μ_i is restricted to this range, the probability of $NN(R_i, q, k_i)$ failing to be evaluated by the RSCM method is estimated to be low (less than 0.004285).

In other words, for this choice of range and samples, the probability that none of the proxy patches are evaluated is less than 1 in 233. This error bound is quite conservative - in practice, the probability of failure would be far smaller.

Even when the RSCM method promotes a proxy patch $NN(R_i, q, k_i)$ as a cluster estimator, there is no precise way of inferring the size of the corresponding cluster in S. However, following the principle of maximum likelihood estimation, the value $c = E[k] \ |S| / |R_i|$ at which $E[k] = k_i$ constitutes a natural estimate of the true cluster size. The smallest cluster size that can be estimated with respect to sample R_i is therefore $(a \ |S|) / |R_i|$.

It should be noted that when the same cluster is detected several times

over several different samples, the estimates of the true cluster size may not agree. Nevertheless in practice, a large RSCONF value will generally turn out to be a reliable indicator of the presence of a cluster, even if the size of the cluster cannot be precisely determined.

< Element Reclassification>

Further in the present invention, by virtue of the proximity of their members to a common query element, clusters produced by the RSCM method tend to be much more cohesive than those produced by agglomerative clustering methods, a desirable trait in the context of text mining. In particular, query clusters are biased towards shapes that are spherical relative to the pairwise distance metric.

Although the solution cluster patch for the RSCM problem as a whole exhibits a high level of mutual association relative to others based at the same query element, the members of such a cluster may or may not be strongly associated with the query element itself. Rather, the query element merely serves as a starting point from which a mutually well-associated neighborhood of the data can be discovered. When the query element is an outlier relative to its associated cluster, or in other situations in which a substantial portion of the reported cluster seems composed of outliers, it may be advantageous to reassess the outer patch elements according to a secondary clustering criterion. Such reassessment allows the discovery of cohesive clusters with less spherical bias.

Many methods may be possible for reclassifying the elements in the vicinity of a query cluster. A pseudo-code description of one such variation appears in Fig. 8. The process described in Fig. 8 is given below:

i) Given the inner k-patch that determined the original query cluster, a ll members of the corresponding outer patch are reassessed according to the actual number of k-nearest neighbors shared with the query element. In particular, every $v \in NN(R, q, \phi(k))$ is ranked according to the con fidence value $CONF(C_q, C_v)$, where $C_q = NN(R, q, k)$ and $C_v = NN(R, v, k)$, from highest to lowest (ties are broken according to distance from q).

ii) The k elements having highest score can be reported as the new, adju sted cluster; alternatively, the entire ranking of the outer patch eleme nts can be reported, and the user left to judge the final cluster member ship. In this way, elements outside the original inner patch yet inside the outer patch are eligible for inclusion in the new cluster, provided they have a high number of original patch members among their nearest ne ighbors.

<Selection of Clusters>

The proposed total clustering strategy, the function PatchCluster constr ucts a query cluster relationship (QCR) graph drawn from a collection of uniform random samples $\{R_0, R_1, R_2, \ldots\}$ such that $R_i \subset R_j$ for all $j < i$ and $|R_i| = ceil(|S|/2^i)$ for $0 \leqq i < \log_2 |S|$. The graph structure de pends on several parameters resembling the confidence and support thresh olds used in association rule generation:

i) (cluster quality) a minimum threshold α on the relative self-confide nce of clusters;

ii) (cluster differentiation) a maximum threshold β on the confidence b etween any two clusters of roughly the same size (drawn from a common sa mple R_i);

iii) (association quality) a minimum threshold γ on the confidence betw een associated clusters (not necessarily drawn from a common sample);

iv) (association scale) a maximum threshold δ on the difference in scal

e between two associated clusters (that is, the difference |i-j|, where R_i and R_j are the samples from which the clusters derive).

Fig. 9 shows a sample pseudo-code description of the Patchcluster method used in the present invention. The basic QCR construction strategy can be summarized as follows:

1.QCR node set:

For each $0 \leqq t < \log_2 |S|$, from the elements of sample R_t, generate a collection of query clusters $QC_t = \{C_1, C_2, \ldots, C_{|Rt|}\}$, with each cluster $C_i = NN(R_t, q_i, k_i)$ based at a different query element of R_t, and $a \leqq |C_i| \leqq b$. Choose the membership of QC_t in greedy fashion from among the available query clusters according to RSCONF values, where $i < j \Rightarrow RSCONF(C_i) \geqq RSCONF(C_j)$, subject to two conditions:

i. (cluster differentiation) $\max \{CONF(C_i,C_j), CONF(C_j,C_i)\} < \beta$ for all $1 \leqq i < j \leqq m_t$.;

ii.(cluster quality) $RSCONF(C_i) \geqq \alpha$ for all $1 \leqq i \leqq |R_t|$.

These clusters become the nodes of the QCR graph at level t.

2.QCR edge set:

For each pair of distinct query clusters $C_i = NN(R_i, q_i, k_i)$ in QC_i and $C_j = NN(R_j, q_j, k_j)$ in QC_j such that $i \leqq j \leqq i+\delta$, insert directed edges (C_i,C_j) and (C_j,C_i) into the QCR graph if $\max \{CONF(C_i,C'_j), CONF(C'_j,C_i)\} \geqq \gamma$, where $C'_j = NN(R_i, q_j, 2^{j-i}k_j)$. Apply the values $CONF(C_i,C'_j)$ and $CONF(C'_j,C_i)$ as weights of edges (C_i,C_j) and (C_j,C_i), respectively.

Each level of the graph can be viewed as a rough slice of the set of clusters, consisting of those with estimated sizes falling within a band depending upon the level, and upon a and b. Within each slice, candidates

are chosen greedily according to their RSCONF values, with new candidate s accepted only if they are sufficiently distinct from previously-accept ed candidates.

In the present invention, although duplicate clusters occurring at a com mon level are eliminated, duplicate clusters are tolerated when they occ ur at different levels. The QCR graph can thus contain any given cluster only a small number of times. The presence of the same cluster at sever al consecutive levels actually improves the connectivity of the structur e, as two query clusters sharing a common concept are likely to be deeme d to overlap, and thereby be connected by an edge. Fig. 10 shows a sampl e pseudo-code listing for eliminating clusters, referred to as the "Pat chcluster method" in the present invention.

By lowering or raising the value of α, users can increase or decrease t he number of cluster nodes appearing in the graph. Raising the value of β also increases the number of clusters; however, this comes at the ris k of individual concepts being shared by more than one cluster from a gi ven sample. Users can vary the value of γ to influence the number of gr aph edges. For the purpose of navigating the clustering results, high gr aph connectivity is desirable. The maximum threshold δ on the differenc e in scale between two associated clusters of the QCR graph should be a small, fixed value, for reasons that will be discussed later.

Another variation of the PatchCluster method involves the control of the number of clusters. As described above, the number of clusters produced is controlled by specifying a threshold α on the relative self-confide nce of the query clusters reported. Instead, the user may be given the o ption of determining the number of clusters for each data sample separat

ely. For a given level t, this can be done by:
i)specifying a minimum threshold α_t on the relative self-confidence of the query clusters to be reported from level t, or
ii)specifying a maximum threshold on the absolute number of query clusters to be reported from level t.

When a threshold on the number of clusters is given, the greedy selection of clusters terminates when the desired numbers of clusters have been obtained, or when all candidates have been considered (whichever occurs first).

In the Patchcluster method, PatchCluster / RSCM parameters may be determined depending on the system to which the above described method or algorithm is implemented. The parameters determined are as follows:

<Inner Patch Size Range>
The inner patch size range [a, b] should be chosen so as to allow arbitrarily-large clusters to be discovered by the method. Although more precise choices of a and b are possible by analyzing the probability of failure (using Chernoff bounds as described earlier), the following general principles apply; parameter a should be large enough to overcome the variation due to small sample sizes. It is recommended that the variable a be no smaller than 20. Parameter b should be chosen such that the ranges of cluster sizes targeted at consecutive levels has substantial overlap. This is achieved when b is roughly 3 times as large as a, or greater.

<Maximum Patch Size>
Also the maximum patch size should be chosen to be as small as possible for reasons of efficiency. However, it should be chosen to be substantia

lly larger than b. The choice ϕ (b) = 2b is ideal; however, the choice ϕ (b) = 1.25b can also give good results. In the best embodiment of the present invention, a = 25, b = 120, and ϕ (k) = min {2k, 150} were prefe rred because satisfactory results were obtained with many data sets.

The maximum threshold b on the confidence between any two clusters from a common sample should be set to roughly 0.4, regardless of the data set . Experimentation showed that overlapping query clusters from a common s ample tend either to overlap nearly completely, or only slightly. The cl ustering produced by the PatchCluster method is relatively insensitive t o the exact choice of b.

<The Threshold δ >

The maximum threshold δ on the difference in scale between two associat ed clusters of the QCR graph should always be a small, fixed value, for several reasons. Large values will lead to graphs in which the largest c lusters would be connected to an overwhelming number of very small clust ers. As a result, the QCR graph would become very difficult for users to navigate. For every query cluster from level 0, a neighborhood of the f orm $NN(R_i, q_j, 2^{\delta} k_j)$ would need to be computed. To ensure scalability, δ must be chosen to be a small constant. The value used in the experiment ation, δ = 4, allowed association edges to be generated between cluster s whose sizes differ by at most a factor of roughly 2^4 to 2^5. This choic e of δ is strongly recommended.

The following parameters should be set by users according to their parti cular demands:

(a)The minimum threshold α on the relative self-confidence of clusters (or alternatively, for each sample level, the minimum cluster relative s

elf-confidence and/or the maximum number of desired query clusters). Values in the range $0.1 \leqq \alpha \leqq 0.2$ are recommended; the smaller the value, the greater the number of clusters.

(b)The minimum threshold γ on the confidence between associated clusters in the QCR graph (not necessarily drawn from a common sample level). Values in the range $0.15 \leqq \gamma \leqq 0.2$ are recommended; the smaller the value, the greater the number of edges of the graph.

(c)The number of keyword labels to be applied to each query cluster.

A further variation of the PatchCluster method is to compute approximate neighborhood patches instead of exact ones. The neighborhood computation performed by the PatchCluster method can be expensive if the number of data elements is large and exact neighborhood information is sought. To improve the efficiency of the method, approximate neighborhood information can be substituted. Similarity search structures such as a SASH can be used to generate this information much faster than sequential search at high levels of accuracy.

A further variation of the PatchCluster method variation is to perform dimensional reduction of the document-keyword vectors and keyword vectors.

The basic PatchCluster method, as described in Fig. 9, when applied to text data, assumes that documents have been modeled as vectors using an appropriate weighting. When the keyword space is large, but the average number of keywords per document is small, distance computations between vectors can be performed efficiently if the vectors are represented implicitly (that is, if only non-zero entries and their positions are stored). However, when the average number of keywords per document is large, di

mensional reduction is often performed in order to limit the cost of distance comparisons. Regardless of the original average number of keywords per document, dimensional reduction techniques such as LSI or COV can be applied to the data before clustering, if desired. The experimental results presented in the Embodiments section show the respective advantages of the use or non-use of dimensional reduction.

Yet another variation of PatchCluster method variation is possible by incorporating QCR graph simplification. The QCR graph produced by the PatchCluster method contains association information for many pairs of clusters. However, this information may sometimes be too dense for users to easily navigate without simplification. Some of the ways in which the graph could reasonably be simplified are:

i) (Elimination of transitive edges between levels.) For example, assume the graph contains cluster nodes $C_1 = NN(R_u, q_1, k_1)$, $C_2 = NN(R_v, q_2, k_2)$, and $C_3 = NN(R_w, q_3, k_3)$, where $u < v < w$, and association edges (C_1, C_2), (C_2,C_3) and (C_1,C_3). Then edge (C_1,C_3) can be hidden from the user, since he or she would still be able to navigate from C_1 to C_2 via (C_1,C_2) and (C_2,C_3).

ii) (Contraction of similar clusters.) If two clusters $C_1 = NN(R_u, q_1, k_1)$ and $C_2 = NN(R_v, q_2, k_2)$ are deemed to be very similar due to sufficiently high values of both $CONF(C_1,C_2)$ and $CONF(C_2,C_1)$, then their respective nodes can be contracted. One of the two nodes is retained and the other is eliminated (the retained cluster node can be chosen in a variety of ways, such as the one with higher RSCONF value, or the one with larger size). Any edges involving the eliminated node are then assigned to the retained node; for example, if C_1 is retained and C_2 is eliminated, then the edge (C_2, C_3) is converted to (C_1,C_3). Any duplicate edges that result would also be eliminated. Of course, other simplification methods

may be adopted in the present invention.

<Graphical User Interface: Cluster Labeling>

In order to provide a useful graphical user interface for displaying sea rched clusters, the problem of query cluster labeling and identification will now be considered in the context of textual data and vector space modeling. Since query clusters lie within a restricted neighborhood of a single query element, it is tempting to use the query as a descriptor o f the cluster, much as in representative-based clustering. However, the query element may not necessarily be the best representative for its clu ster; indeed, it may be the case that no individual element of the clust er adequately describes the whole.

One common way of assigning labels to a cluster is to use a ranked list of terms that occur most frequently within the documents of the cluster, in accordance with the term weighting strategy used in the document vec tor model. Each term can be given a score equal to the sum (or equivalen tly the average) of the corresponding term weights over all document vec tors of the clusters; a predetermined number of terms achieving the high est scores can be ranked and presented to the user.

If dimensional reduction techniques such as COV or LSI are being used, t he original unreduced document vectors may no longer be available, or ma y be expensive to store and retrieve. Nevertheless, meaningful term list s can still be extracted even without the original vectors. Note first t hat the i-th term can be associated with a unit vector $z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,d})$ in the original document space, such that $z_{i,j} = 1$, if $i = j$, and $z_{i,j} = 0$ otherwise. Now, let μ be the average of the document vec tors belonging to the query cluster NN(R, q, k). Using this notation, th

e score for the i-th term can be expressed simply as $z_i \cdot \mu$. However, since $\| z_i \| = 1$ and μ is a constant, ranking the terms according to these scores is equivalent to ranking them according to the measure as below:

$$z_i \cdot \mu / \| \mu \| = \text{cosangle}(z_i, \mu) = \cos \theta_i,$$

where θ_i represents the angle between vectors z_i and μ.

With dimensional reduction, the pairwise distance cosangle(v, w) between vectors v and w of the original space is approximated by cosangle(v', w'), where v' and w' are the respective equivalents of v and w in the reduced dimensional space. Hence we could approximate cosangle(z_i, μ) by cosangle(z'_i, μ'), where z'_i and μ' are the reduced-dimensional counterparts of vectors z_i and μ, respectively. The value cosangle(z'_i, μ') can in turn be approximated by cosangle(z'_i, μ''), where μ'' is the average of the reduced-dimensional vectors of the query cluster. Provided that the vectors z'_i have been precomputed for all $1 \leqq i \leqq d$, a ranked set of terms can be efficiently generated by means of a nearest-neighbor search based on μ'' over the collection of reduced-dimensional attribute vectors. As d is typically quite small, the cost of such a search is negligible compared to the cost of generating the cluster itself.

The reduced-dimensional cluster labeling method can be summarized as follows:

i) For all $1 \leqq i \leqq N$, precompute the reduced-dimensional attribute vector $z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,d})$ for the i-th attribute. Let W be the se

t of reduced-dimensional attribute vectors.

ii) Compute μ'' = $S_v \in NN(R, q, k)$ v, where v and q are taken to be reduced-dimensional data vectors.

iii) If λ is the desired number of labels for the cluster, compute the λ-nearest-neighbors of μ'' in W, according to decreasing values of the cosangle measure.

iv) Report the attributes corresponding to the ranked list of λ neighbors as the cluster labels. Optionally, the values of cosangle themselves can be displayed to the user. Also optionally, approximate nearest neighbors can be used as produced using a SASH or other similarity search method.

Part II A System for Information Retrieval

Fig. 10 shows a system to which the algorithm of the present invention is implemented. As shown in Fig. 10, the system generally comprises a computer 10, a display device 12, and a input device such as a keyboard 14 and a pointer device such as an mouse 16 such that a user may input a query for information retrieval according to the present invention. The computer 10 also manages a database 18 for storing documents to be searched. The computer may add new documents to the database 18 and retrieve stored documents therefrom. The computer 10 may be connected to communication lines 20 such as LAN or WAN or Internet such as Ethernet (Trade Mark), an optical communication, or ADSL with suitable communication protocols through a hub or router 22.

When the communication line 20 is assumed to be LAN/ WAN and/or Internet locally interconnecting sites of an enterprise, the computer 10 may be a server to which inputted queries from clients and/or users are transmitted to execute information retrieval. The server computer 10 retrieves

documents with respect to the received query by the algorithm of the pre sent invention and returns the retrieved results to the clients that iss ued the query. Of course, the present invention may provide the above in formation retrieval through the Internet as charged information services to registered clients. Alternatively, the computer 10 may be a stand-al one system suitably tuned for a particular usage.

Fig. 11 shows detailed functional blocks implemented in the computer 10. The computer 10 generally comprises a vector generation part 24, a SASH generation part 36, a confidence determination part 38 for creating the SCONF list, and a patch definition part 26. The vector generation part 24 executes vector generation using a keyword list or predetermined rule s from the documents stored in a database 18, and stores generated docum ent-keyword vectors in an appropriate storage area such as a memory or a database with adequate links or references to the corresponding documen ts. The SASH generation part 36 and the patch definition part 26 constit ute the neighborhood patch generation part 34 according to the present i nvention.

The SASH generation part 36 constructs the SASH structure using the algo rithm shown in Fig. 2 and the generated SASH structure is stored in the memory area 30 for the processing described hereinafter in detail. The S ASH is made available to a confidence determination part 38 to compute c onfidence values such as CONF, SCONF, and RSCONF so as to generate a SCO NF list according to the above described algorithm. The generated patch data and the confidence values are stored in a hard disk 32, as shown in Fig. 7.

The query vector generation part 46 accepts search conditions and query

keywords and creates a corresponding query vector, and stores the genera ted query vector in an adequate memory area. The query may be of two typ es; one is to extract cluster structures already computed and stored in the database 32, and the other is to retrieve cluster structures that ma y not yet have been computed and stored. The user input query vector is first transmitted to a retrieval part 40. In the described embodiment, t he retrieval part analyses the query type. If the query instructs the re trieval part 40 to retrieve cluster structures already computed and stor ed, the query is performed on the SASH structure stored in the memory ar ea 30, and the queried patch generation part 44 transmits the retrieved data to the cluster estimation part 28. The cluster estimation part 28 i nvokes the patch data and associated SCONF list from the hard disk 32 up on receiving the retrieved data, and performs cluster estimation using i ntra-cluster confidences SCONF and RSCONF, and inter-cluster confidences CONF, respectively. The nodes used in the queried patch generation part 44 may be an arbitrarily selected node or a node retrieved by the user input query.

The derived cluster data are transmitted to a GUI data generation part 4 2 to construct data for graphically presenting the cluster graph structu re on a display screen of a display part (not shown). Many display embod iments of the cluster graph structure are possible in the present invent ion. One representative embodiment is to align the clusters horizontally along with significant keywords (for example, the largest numeral value s) included in the clusters while aligning the clusters vertically with estimated cluster size. When such display is provided on the display scr een, the GUI data generation part 42 may sort the cluster data from the patch repository 32, and store the sorted data in an adequate memory are a therein such as an display buffer (not shown), or elsewhere in the com

puter 10.

In an specific embodiment of the present invention, when the retrieval part 40 determines that the query instructs the retrieval of clusters that have not already been computed and stored, the retrieval part 40 invokes the SASH data 30, and retrieves the appropriate node vectors of the SASH by computing similarities between the document-keyword vectors and the query vectors. The retrieved data vectors are then themselves used as queries within the SASH data 30, to obtain a list of similar node vectors for every vector retrieved by the original query. Each list of node vectors is sent to the patch definition part 26 and thence to the confidence determination part 38 to produce patches, which may then be added to the patch repository 32. The retrieved patches are then transmitted to the cluster estimation part 28 together with their corresponding SCONF lists to estimate the cluster comprising nodes retrieved in the original query, and the computed cluster data are transmitted to the GUI data generation part 42 for graphical presentation of the queried results.

The GUI data generation part 42 may transmits sorted cluster data to a display device (not shown) directly connected to the computer 10 to display the searched cluster data on a display screen. Alternatively, when the system provides the searched results via the Internet using a browser software, the GUI data generation part 42 generates graphical data of the interrelation of the clusters in a format suitable to the browser software, such as an HTML format.

Part III. Practical Scenarios for Executing Invention

<Scenario A - Total Clustering of Nodes in Database>

Fig. 12 shows a flowchart for Scenario A for executing total clustering

of the nodes stored in the database. The algorithm of Scenario A first l oads the document and the keyword data in the step S40 and proceeds to t he step S42 to generate document-keyword vectors and keyword lists. The algorithm executes in the step S46 dimension reduction using LSI or COV as described before. Then the process of Scenario A creates in the step S46 a SASH of the dimension reduced document-keyword vectors according t o the process described in Fig. 2. The data structures generated accordi ng to the algorithm shown in Fig. 12 are shown in Fig. 13 according to t he stepwise execution of the algorithm.

Once the SASH structure has been constructed, a similarity query is perf ormed based on each of its elements, thereby generating one patch for ea ch document, as shown in Fig. 14(a). The algorithm of Scenario A then co mputes in the step S48 the optimum patch sizes and RSCONF values for eac h patch as shown in Fig. 14 (b), and then the patches are sorted with re spect to their RSCONF values as shown in Fig. 14(c).

Again referring to Fig. 12, the algorithm of Scenario A proceeds to the step S50 to select, at each SASH level, a collection of patches for whic h all inter-patch association confidences at that level are less than β = 0.4. Then those patches with RSCONF values larger than or equal to α = 0.15 are further selected to determine the clusters, in the step S52. The data structures relevant to the steps S46-S52 are shown in Fig. 15.

Next, the algorithm of Scenario A proceeds to the step S54 to create con nections between the clusters having the association confidence values l arger than or equal to the predetermined threshold γ. This data structu re is shown in Fig. 16. These results of connection together with the cl uster labels and corresponding keywords are provided graphically in the

step S56 as a graphical representation such as that shown in Fig. 17. In Fig. 17, a portion of a cluster graph produced according to Scenario A (on an earlier run with γ = 0.2) is shown in Fig. 17 for the case in which COV dimensional reduction was used. In the figure, cluster nodes (shown as ovals) are marked with a pair of numbers x / y, where x indicat es the estimated size of the cluster and y indicates the associated samp le patch size. Keyword labels are shown for each cluster - boxes have be en drawn around those connected subsets of clusters sharing identical la bel sets (with perhaps minor differences in the label ordering). The clu ster corresponding to the node marked 106 / 53 is shown in Fig. 17. This cluster is particularly interesting, as it consists of news articles in the intersection of two larger sets of clusters, involving canyons and their development and conservation issues on the one hand, and garbage d umps and landfills on the other.

The detailed procedure included in the processes shown in Fig. 12 are de scribed as below:

i) Model the subset of M documents as vectors, using (for example) the b inary model or TF-IDF weighting. The dimension of these vectors is N, th e number of attributes of the data set.

ii) As a further example, apply dimensional reduction of the set of vect ors to a number significantly smaller than N (typically 200 to 300), usi ng (for example) the COV or LSI reduced-dimensional technique. If dimens ional reduction is chosen, then also generate a set of reduced-dimension al attribute vectors.

iii) Construct the SASH structure for handling k-nearest-neighbor querie s. Set the random sample $R_t = S_t \cup S_{t+1} \cup \ldots \cup S_h$, where S_t is the t -th SASH level for $0 \leqq t \leqq h$ (here, S_0 is taken to be the bottom SASH level).

iv) For all $0 \leqq t \leqq h$, for each element $v \in S_t$, compute and store an approximate m-nearest-neighbor list (m-patch) $NN(R_t, v, m)$ for that element, where $m = \phi(b)$.

v) Compute a set of query clusters and a cluster structure graph as outlined in Fig.16.

vi) When the dimension reduction is performed, for each query cluster of the set, generate a set of attribute keywords from the reduced-dimensional document vectors that constitute the cluster.

vii) Make the resulting set of clusters, their sizes and labels, and cluster structure graph available to the user for browsing, using a suitable user interface.

<Scenario B - Individual Clusters; Query Search>

In Scenario B, the same process as in Scenario A may be used to generate a SASH. The subsequent essential steps are shown in Fig. 18, and the data structures generated by the process of Scenario B are shown in Fig. 19 and Fig. 20. As shown in Fig. 18, the process of Scenario B generates the SASH structure in the step S60, and proceeds to the step S62 to receive a user input query q together with a target cluster size k, and stores them in an adequate memory space. Then the nodes in SASH are retrieved with respect to the query using the SASH structure in the step S64. In the step S64, the SASH is queries to produce one neighborhood patch for the query element q with respect to each of the random samples R_t, for all $0 \leqq t \leqq h$. Then the process continues to the step S66 to compute RSCONF and to solve the RSCM problem with respect to the user input query q, for every random sample. For each sample, a cluster is thereby produced. The process of Scenario B then provides labels, keywords representing these clusters, in the step S68. The data structures obtained from the step S64 to the step S68 are shown in Fig. 20.

The details of the procedures in Scenario B are described below:

2-i) Repeat the procedures of Scenario A from i to iii.

2-ii) Prompt the user for a query element q (not necessarily a data element), and a target cluster size k.

2-iii) Compute $t_a = \max \{t \mid k / 2^t \geqq a\}$ and $t_b = \min \{t \mid k / 2^t \leqq b\}$. For all $t_b \leqq t \leqq t_a$, compute $NN(R_t, q, m)$, where $m = \phi(b)$. For all $v \in NN(R_t, q, m)$, compute $NN(R_t, v, m)$.

2-iv) For all $t_b \leqq t \leqq t_a$, find solutions $k(q,t)$ to the RSCM problems for q with respect to R_t.

2-v) For all $t_b \leqq t \leqq t_a$, generate a set of attribute keywords from the reduced-dimensional document vectors that constitute the query cluster $NN(R_t, q, k(q,t))$. The procedure has been described in Fig. 14.

2-vi) Display the resulting set of clusters, their sizes, their corresponding m-patch SCONF profiles, and their cluster labels to the user.

[Examples]

To examine the present invention, the method of the present invention was implemented as two scenarios as described above. Both scenarios were examined for the publicly-available L.A. Times news document database available as part of the TREC-9 text retrieval competition. The database consists of M = 127,742 documents, from which N = 6590 keywords (attributes) were extracted as the attribute set. To examine effectiveness and general applicability, the database was subjected to two procedures with and without the dimension reduction (under COV). The implementation conditions were as follows:

(a) TF-IDF term weighting on 6590 attributes.

(b) COV dimensional reduction (from 6590 down to 200 dimensions) in one

set of experiments, and no dimensional reduction in another.
(c) For document nearest-neighbor searches, a SASH with default settings (node parent capacity p = 4 and node child capacity c = 16).
(d) For attribute vector nearest-neighbor searches, a SASH for reduced-dimensional attribute vectors with default values (node parent capacity p = 4 and node child capacity c = 16).

For each scenario, it was assumed that parameters ϕ, a, b, β, and δ were set by the system administrator, as well as any parameters associated with dimensional reduction (such as the reduced dimension d) or approximate similarity search.

The experimental conditions are as follows:
(a) The choice of patch range delimiters a = 25, b = 120, and $\phi(k) = \min\{2k, 150\}$.
(b) For document nearest-neighbor searches, the use of a time scaling factor
$\mu' = 1.25$
$\mu = 1.25\ \phi(b)$
influencing the accuracy of the approximation. With every search, μ' neighbors are produced, of which the closest m are used (larger values of μ' require longer search times but lead to more accurate results).
(c) A minimum threshold of $\alpha = 0.15$ on the relative self-confidence of clusters.
(d) A maximum threshold of $\beta = 0.4$ on the confidence between any two clusters from a common sample.
(e) A minimum threshold of $\gamma = 0.15$ on the confidence between associated clusters in the QCR graph (not necessarily drawn from a common sample level).

(f) A maximum threshold of $\delta = 4$ on the difference in scale between two associated clusters of the QCR graph.

The computation algorithm was written using Java (JDK1.3) and the computation hardware was an IBM Intellistation E Pro (Trade Mark) with 1GHz processor speed and 512Mb main memory running the Windows 2000 (Trade Mark) operating system.

2-1. Execution time and storage costs

Although at first glance it would seem that RSCONF values are expensive to compute, with a careful implementation the costs can be kept reasonably low. This is achieved through the efficient computation of a profile of values of SCONF(NN(R, q, k)) for k ranging from 1 to ϕ(b). Plots of patch profiles also provide an effective visual indication of the varying degrees of association within the neighborhood of a query element.

The following tables list the time and space costs associated with Scenario A. Time was measured in terms of real seconds of computation, beginning once reduced-dimensional document and attribute vectors had been loaded into main memory, and ending with the computation of a full set of clusters and their cluster structure graph. The time cost for clustering and graph construction assumes that all nearest-neighbor patches have already been precomputed.

Table 1

STORAGE COSTS (Mb) - Reduced Dimensional Case	
Document SASH Storage	30.1
Keyword SASH Storage	1.6
NN Patch Storage	161.6
Reduced-Dimensional Document Storage	204.4
Reduced-Dimensional Keyword Storage	5.3
Total Storage	403

Table 2

TIME COSTS	No Dim-Reduction	COV Dim-Reduction
Document SASH Build Time (s)	460.7	898.8
Keyword SASH Build Time (s)	--	26.6
Total NN Precompution Time (s)	7,742.9	13,854.6
Clustering and Graph Construction Time (s)	126.2	81.8
Total Time (s)	8,329.8	14,861.8
Total Time (hr)	2.3	4.1

2-2. Approximate nearest neighbor computation

The following table shows the average cost of finding approximate m-nearest-neighbor lists from a full set of M documents, taken over 100 randomly-chosen SASH queries of size m' . For comparison purposes, exact queries were also performed using sequential search, and the average accuracy of the SASH queries was computed (measured as the proportion of true nearest neighbors in the reported lists). Using these values, one can determine the cost of producing a single query cluster directly as per Scenario B. These latter estimates assume the use of the document SASH without precomputed nearest-neighbor information.

Table 3

SASH Performance	No Dim-Reduction	COV Dim-Reduction
Avg SASH Query Dist Computations	3,039.03	2,714.57
Average SASH Query Time (ms)	38.85	70.74
Average SASH Query Accuracy (%)	62.93	94.27
Exact NN Query Dist Computations	127,742	
Exact NN Query Time (ms)	1,139.19	2,732.5
Single Query Cluster Dist Comps ($x10^5$)	4.59	4.07
Single Query Cluster Time (s)	5.87	10.68

2-3. Full query clustering

An example of a patch profile is illustrated in Fig. 21, for the case in

which COV dimensional reduction was used. The profile is associated wit h a cluster produced by the Scenario A method.

The numbers of clusters produced under Scenario A with and with out the dimension reduction are listed in the table below.

Table 4

Estimated Cluster Size (low - high)	No Dim-Reduction	COV Dim-Reduction
6400 - 30720	1	1
3200 - 15360	1	2
1600 - 7680	8	8
800 - 3840	15	25
400 - 1920	32	50
200 - 960	70	84
100 - 480	206	135
50 - 240	405	216
25 - 120	760	356

The dimensional-reduction variant finds fewer minor clusters compared to the basic variant, but more larger clusters. Experimentation also revea led that the dimensional-reduction variant produced query cluster graphs with richer interconnections, and was better able to resolve keyword po lysemies.

The method according to the present invention may be implemented as a co mputer executable program, and the computer program according to the pre sent invention may be written in a language such as the C language, the C++ language, Java (trade mark), or any other object-oriented language. The program according to the present invention may be stored in a storag e medium such as a floppy disk (trade mark), a magnetic tape, a hard dis k, a CD-ROM, a DVD, a magneto-optic disk or the like whereto data may be written and wherefrom data may be read which is readable by a computer.

[Effect of Invention]

Improvement 1: MINOR CLUSTERS

The proposed clustering method is able to efficiently detect well-associated and well-differentiated clusters of sizes as low as 0.05% of the database, on an ordinary computer. The methods require no a priori assumptions concerning the number of clusters in the set. The methods also allow clusters to be generated taking only local influences into account. Overlapping clusters are also permitted. These features allow minor clusters to be discovered in a way that is impractical or even impossible for traditional methods.

Improvement 2: QUERY-BASED CLUSTERING

The proposed clustering method can generate meaningful major and minor clusters in the vicinity of a query efficiently, without paying the excessive cost of computing a full clustering of the set. To the best of my knowledge, this is the first practical method for doing so for large text databases.

Improvement 3: AUTOMATIC DETERMINATION of CLUSTER RELATIONSHIPS

Very few clustering methods allow for the possibility of overlapping clusters. The proposed method uses cluster overlap to establish correspondences between clusters, and thereby produce a "cluster map" or graph of related concepts that can be navigated by the user. Unlike concept hierarchies, the relationships are established among groups of data elements themselves, rather than by classifications within the attribute space. Organization according to overlapping clusters of data elements allows for much more flexibility in the concepts that can be represented - in particular, minor clusters in the intersection of two or more major clusters can be discovered using the proposed method.

Improvement 4: AUTOMATIC ASSESSMENT of CLUSTER QUALITY

RSCONF values and patch profiles are techniques that not only serve to identify and compare clusters, they are also the means by which users can assess the level of association within a cluster, and its differentiation with the elements in its vicinity. Patch profiles can effectively complement existing spatial representation methods for the visualization of higher-dimensional text clusters.

Improvement 5: DEPENDENCE on KNOWLEDGE of the DATA DISTRIBUTION

Unlike most partition-based algorithms, the proposed query-based clustering method does not require previous knowledge or assumptions regarding the distribution of the data - it does not matter whether the data is uniformly distributed or has great variations in distribution. This applies even as regards the generation of nearest-neighbor lists, in that the SASH also has this feature.

Improvement 6: SCALABILITY

When a SASH structure is used for approximate similarity queries, the asymptotic time required by PatchCluster for a total clustering of data set S is in $O(|S| \log_2 |S| + c^2)$, where c is the number of clusters produced (typically much smaller than |S|). The former term covers the cost of producing profiles and ranking candidate query clusters according to their RSCONF values. The elimination of duplicate clusters and the generation of graph edges can all be performed in $O(|S| + c \log_2 |S| + c^2)$ time.

The bottleneck in the construction of a query cluster graph lies in the precomputation of nearest-neighbor patches. However, the clustering meth

od does not require perfectly-accurate nearest-neighbor lists in order t o detect approximate cluster boundaries and overlaps. It is far more cos t effective to use one of the emerging techniques, such as the SASH, for fast generation of approximately-correct nearest-neighbor lists instead . For the L.A. Times news article data set using COV dimensional reducti on, the SASH offers speedups of roughly 40 times over sequential search at almost 95% accuracy. The asymptotic complexity of precomputing patche s is dominated by the total cost of the SASH operations, which is in $O(|S| \log_2 |S|)$.

Hereinabove, the present invention has been explained using particular e mbodiments depicted in the drawings. Of course, it is appreciated by a p erson skilled in the art that many alternative embodiments, modification s, and/or additions to the disclosed embodiments may be possible and the refore, the true scope of the present invention should be determined in accordance with the claims herewith.

[Brief Explanation of Drawings]

[Fig. 1] A flowchart of the method for constructing data structures acco rding to the present invention.

[Fig. 2] A simplified flowchart of the process for constructing the SASH structure.

[Fig. 3] A schematic construction of the SASH with patch structures.

[Fig. 4] A sample diagram of the patches according to the present invent ion.

[Fig. 5] A representative example of the computation of the confidence f unction CONF.

[Fig. 6] A sample pseudo-code listing for the computation of SCONFL.

[Fig. 7] An illustration of the structure of patch and self-confidence s

torage.

[Fig. 8] A sample pseudo-code listing for the refinement of patch profil es.

[Fig. 9] A sample pseudo-code listing for PatchCluster (including patch ranking and selection).

[Fig. 10] A schematic block diagram of a computer system typically used in the present invention.

[Fig. 11] A schematic function block diagram of the computer system acco rding to the present invention.

[Fig. 12] A flowchart of the process for generating the clusters and the ir interrelationship graph (Scenario A).

[Fig. 13] A graphical representation of the data structures relevant to the process of Scenario A shown in Fig. 12.

[Fig. 14] A graphical representation of the data structures relevant to the process of Scenario A shown in Fig. 12.

[Fig. 15] A graphical representation of the data structures relevant to the process of Scenario A shown in Fig. 12.

[Fig. 16] A graphical representation of the cluster interrelationship gr aph.

[Fig. 17] A sample graphical presentation of the interrelationship struc ture of clusters.

[Fig. 18] A flowchart of the process for generating clusters based at a single query element (Scenario B).

[Fig. 19] A graphical representation of the data structures relevant to the process of Scenario B shown in Fig. 18..

[Fig. 20] A graphical representation of the data structures relevant to the process of Scenario B shown in Fig. 18.

[Fig. 21] A plot of a profile of SCONF values versus estimated cluster s ize.

[Explanation of numerals]

10 … computer

12 … display device

14 … keyboard

16 … mouse

18 … database

20 … communication line

22 … hub/router

24 … document vector generation part

26 … patch definition part

28 … cluster estimation part

30 … memory or database (SASH storage)

32 … hard disk

34 … neighborhood patch generation part

36 … SASH generation part

38 … confidence determination part

40 … retrieval part

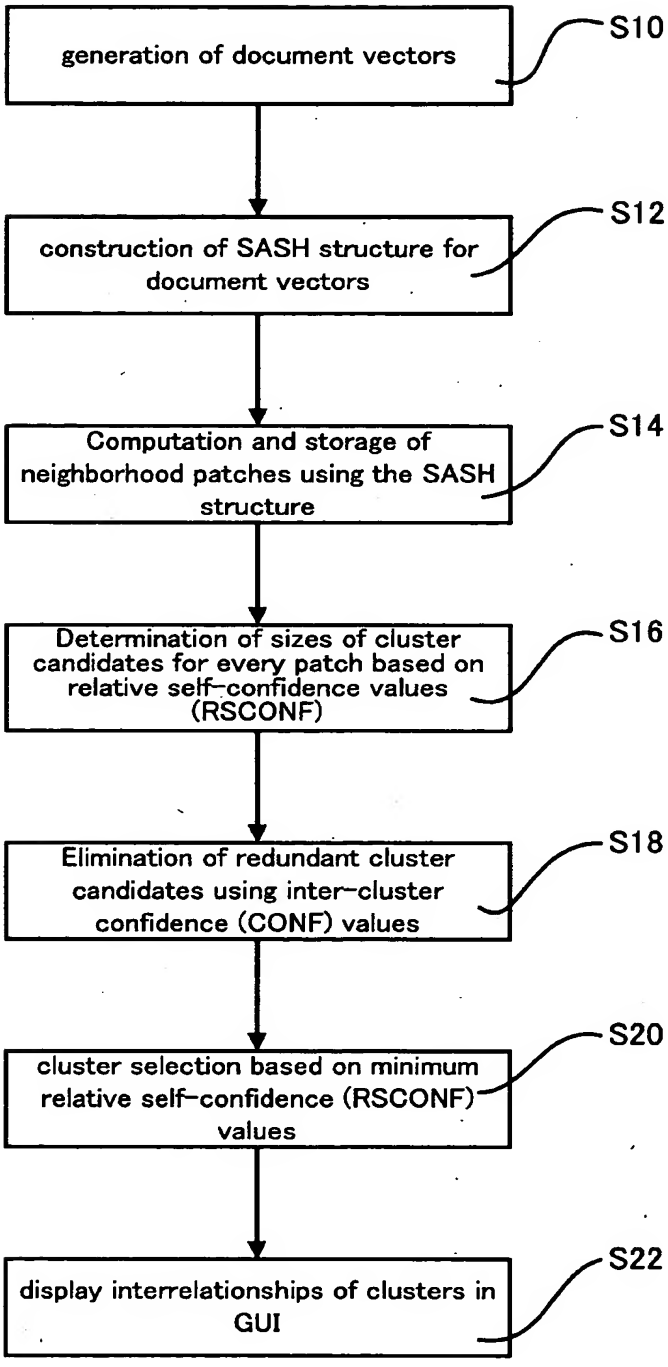
42 … GUI data generation part

44 … queried patch generation part

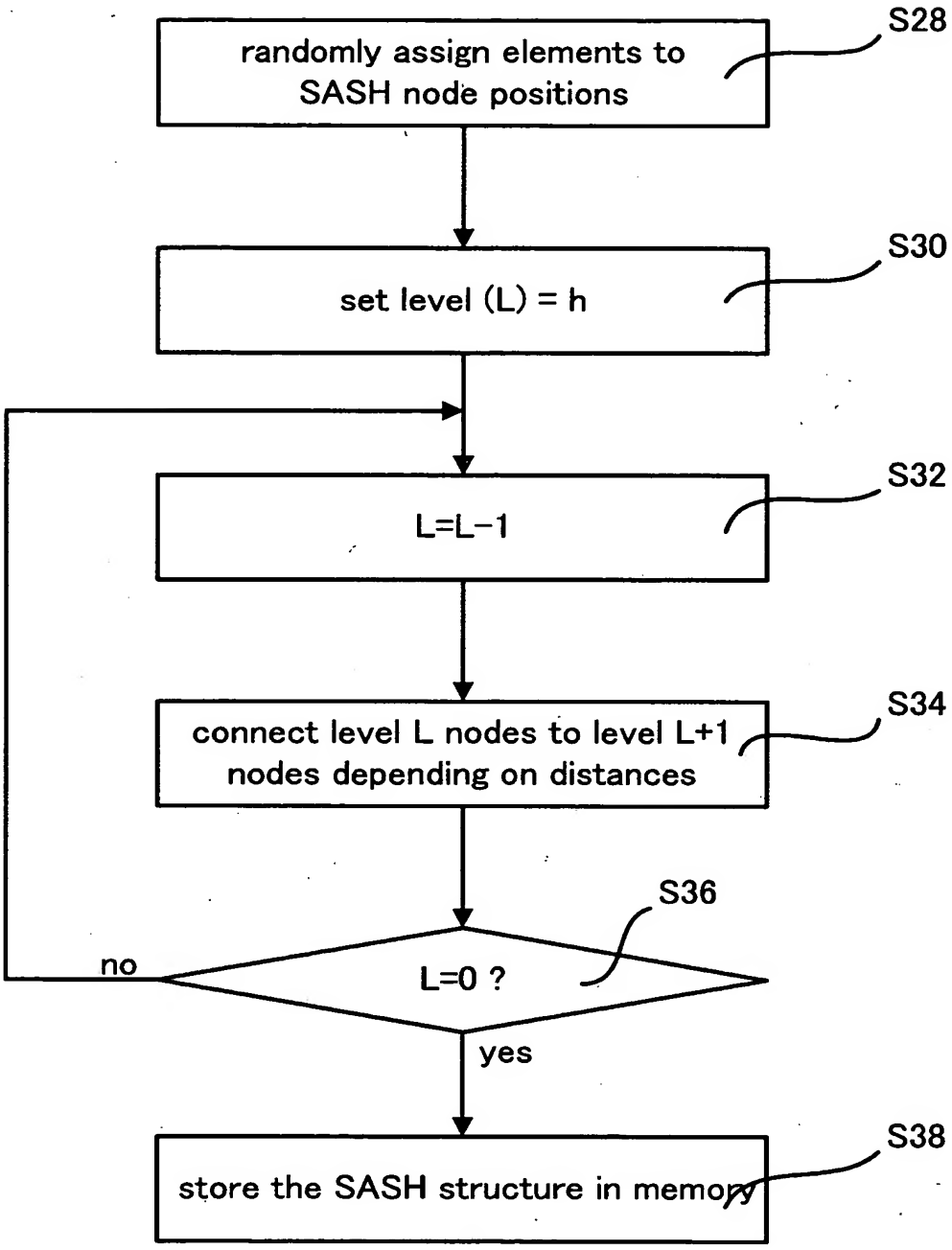
46 … query vector generation part

【書類名】　外国語図面

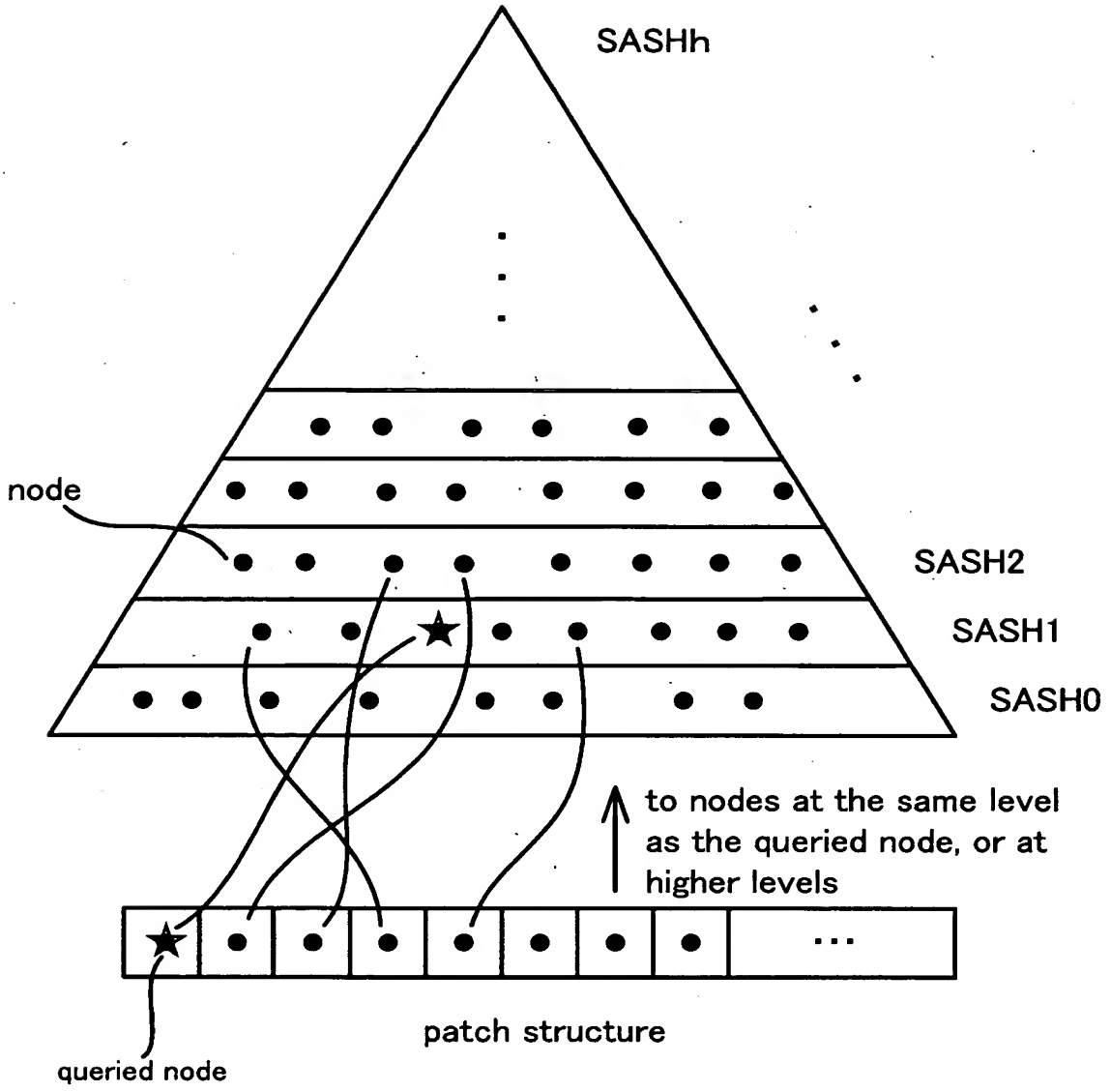
【Fig. 1】



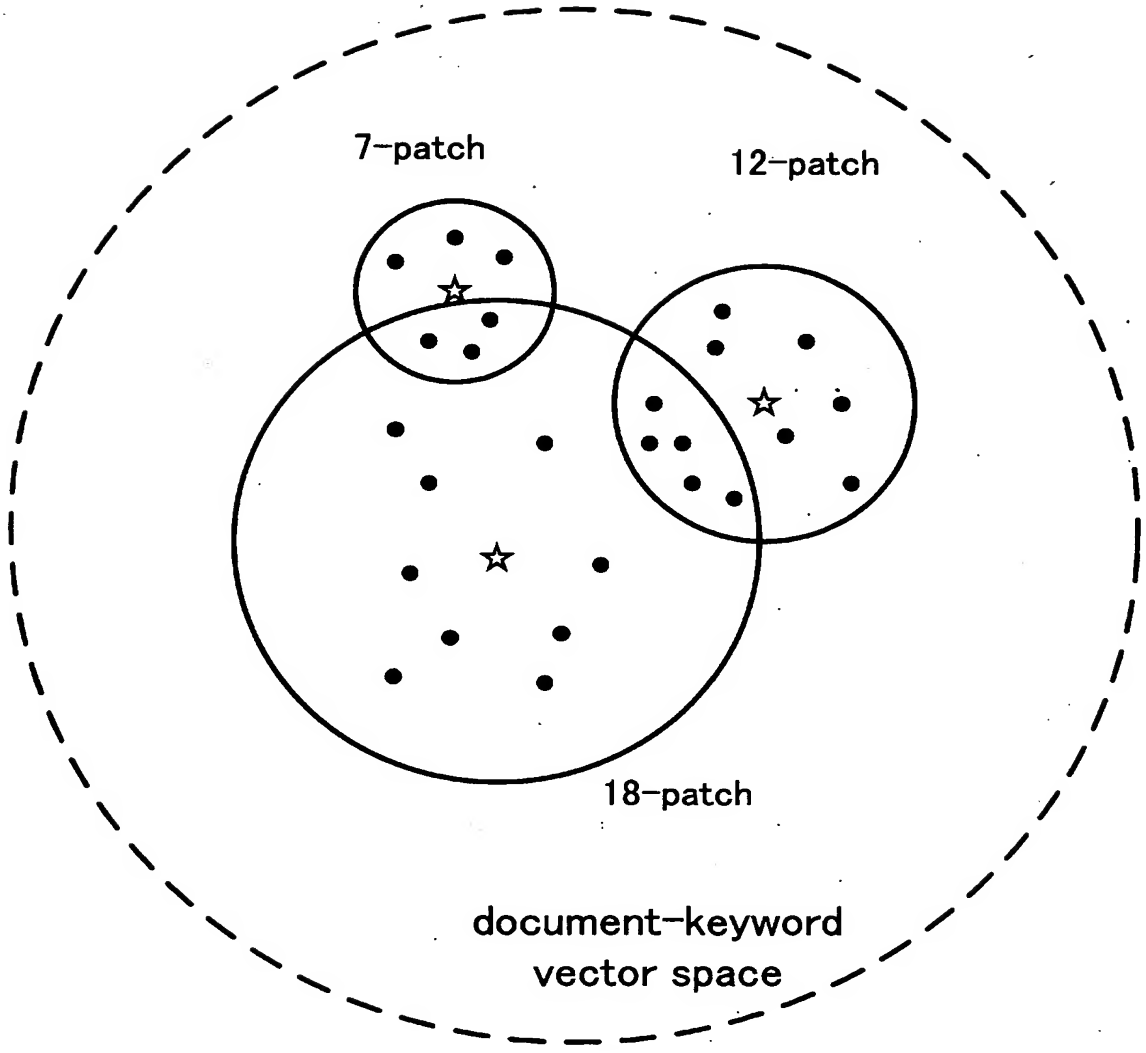
【Fig. 2】


randomly assign elements to SASH node positions
S28
set level (L) = h
S30
L=L−1
S32
connect level L nodes to level L+1 nodes depending on distances
S34
L=0 ?
S36
no
yes
store the SASH structure in memory
S38

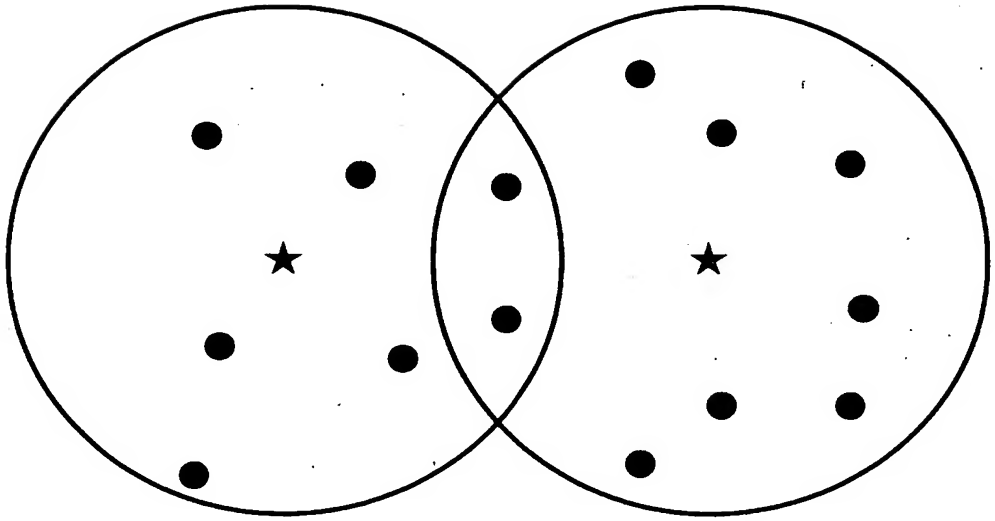
【Fig.3】


SASHh
node
SASH2
SASH1
SASH0
to nodes at the same level
as the queried node, or at
higher levels
patch structure
queried node

【Fig. 4】



【Fig. 5】



$CONF(C_i, C_j) = 2/8 = 25\%$

$CONF(C_j, C_i) = 2/10 = 20\%$

【Fig. 6】

```
Profile (query q; maximum patch size m): SCONF list SCONFL
    {Let QNL be the m-patch prec mputed for query q.}
    {Let NNL be a list of the m-patches precomputed for every element of QNL.}
    {Initially, w.count = ∅ is assumed for every element v in the data set.}
1.  score ← 0;
    {Initially, no query neighbors are in the current patch.}
    for i = 1 to m do
2.      QNL[i].count ← 0;
    end for
    for i = 1 to m do
        {Retrieve the number of times QNL[i] has been encountered as an external neighbor so far.}
3.      score ← score + QNL[i].count;
        {Indicate that henceforth QNL[i] is in the current i-patch.}
4.      QNL[i].count ← present;
        for j = 1 to i − 1 do
5.          w ← NNL[j, i];
            if w.count = present then
6.              score ← score + 1;
            else if w.count ≥ 0 then
7.              w.count ← w.count + 1;
            end if
8.          w ← NNL[i, j];
            if w.count = present then
9.              score ← score + 1;
            else if w.count ≥ 0 then
10.             w.count ← w.count + 1;
            end if
        end for
11.     w ← NNL[i, i];
        if w.count = present then
12.         score ← score + 1;
        else if w.count ≥ 0 then
13.         w.count ← w.count + 1;
        end if
14.     SCONFL[i] = score/i²;
    end for
    {Reset the counts to their default value.}
    for i = 1 to m do
15.     QNL[i].count ← ∅;
    end for
```

【Fig. 7】

Item q_i	SASH level	patch
i=n−1	0	$NN(R_{0,q(n-1)},m)$
i=n−2	0	$NN(R_{0,q(n-2)},m)$
i=n−3	0	$NN(R_{0,q(n-3)},m)$
・ ・ ・	・ ・ ・	・ ・ ・
i=n/2−1	1	$NN(R_{1,q(n/2-1)},m)$
・ ・ ・	・ ・ ・	・ ・ ・
i=n/4−1	2	$NN(R_{2,q(n/4-1)},m)$
・ ・ ・	・ ・ ・	・ ・ ・
i=0	h	$NN(R_{h,q(0)},m)$

【Fig. 8】

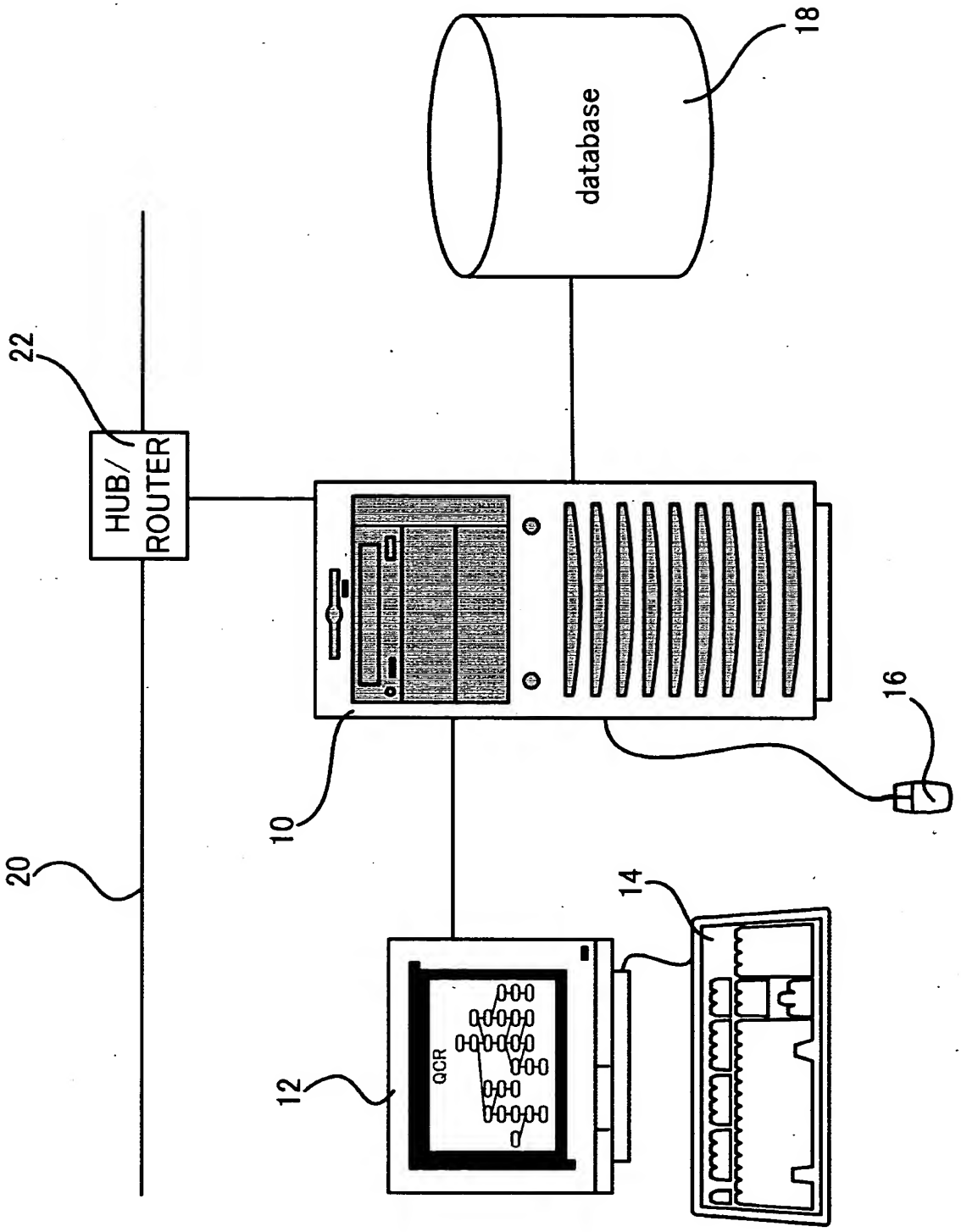
```
RefineProfile (query q;
                inner patch size k_I;
                outer patch size k_O): reordered query k_I-patch RQNL
  {Let QNL be the k_O-patch precomputed for query q.}
  {Let NNL be a list of the k_O-patches precomputed for every element of QNL.}
  {Initially, v.inpatch = false is assumed for every element v in the data set.}
  {Identify the inner patch members.}
  for i = 1 to k_I do
1.    QNL[i].inpatch ← true;
  end for
  {Initialize the confidence value CONFc of every patch element to zero.}
  for i = 1 to k_O do
2.    CONFc[i] ← 0;
  end for
  {For each element of the outer patch, count the number of elements
     of their k-nearest-neighbor sets shared with that of q.}
  for i = 1 to k_O do
     for j = 1 to k_I do
3.      w ← NNL[i, j];
        if w.inpatch = true then
4.            CONFc[i] ← CONFc[i] + 1;
        end if
     end for
5.   CONFc[i] ← CONFc[i]/k_O;
  end for
  {Reorder the outer patch elements according to their confidence values, from highest to lowest.}
6. RQNL ← sort(QNL, CONFc, k_O);
  {Reset the patch membership indicators to their default values.}
  for i = 1 to k_I do
7.    QNL[i].inpatch ← false;
  end for
```

【Fig. 9】

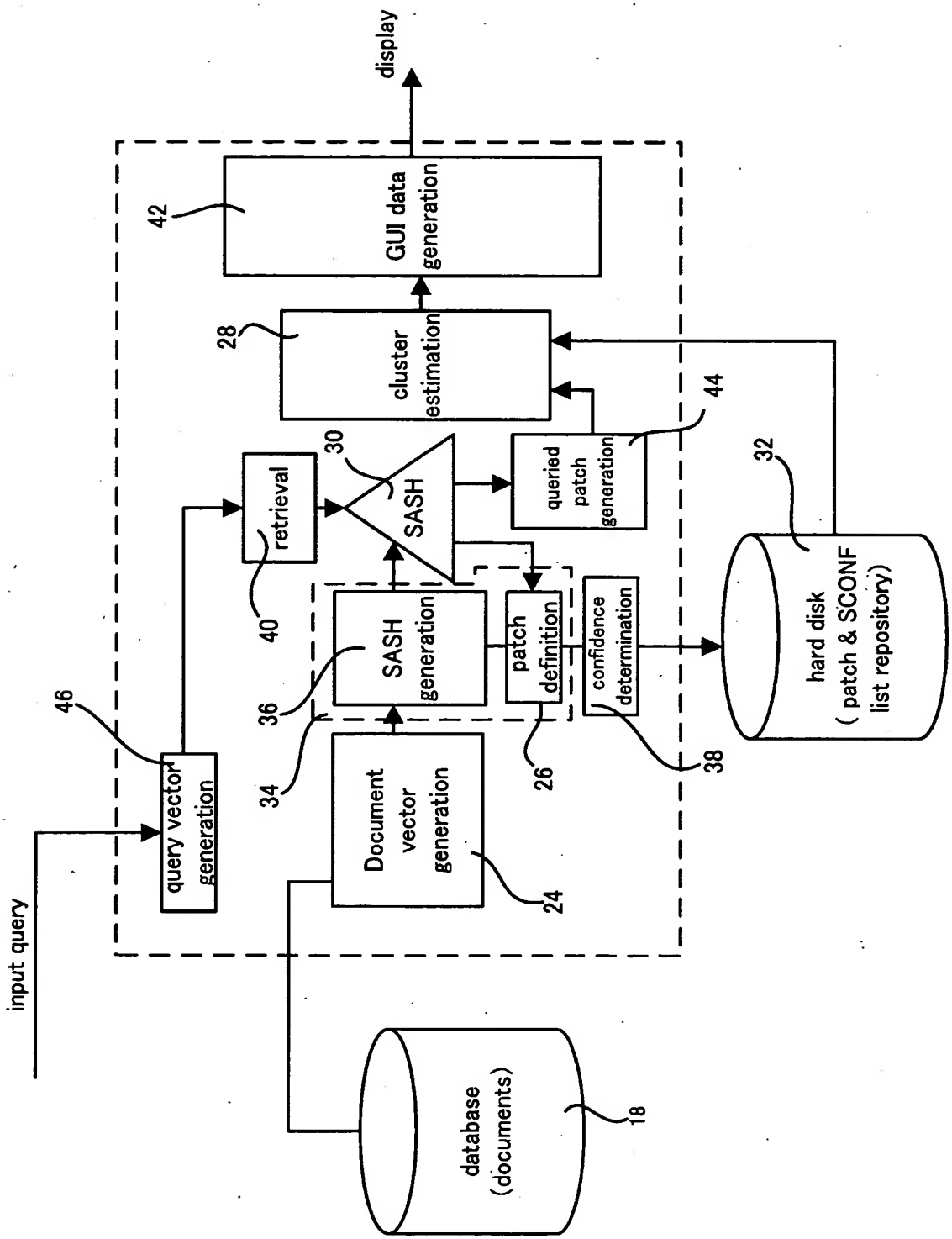
PatchCluster (data set S;
RSCM parameters a, b, $m = \varphi(b)$;
Thresholds α, β, γ, δ): query cluster graph G

1. Randomly partition the set S into subsets S_t of approximate size $\frac{|S|}{2^t}$, for $0 \leq t \leq h = \lfloor \log_2 |S| \rfloor$.
2. For all $0 \leq t \leq h$ do:
   (a) For every element $v \in S_t$, compute nearest-neighbor patches $\mathbf{NN}(R_t, v, m)$, where $R_t = \bigcup_{i \geq t} S_i$.
   (b) For each element $v_{t,i} \in S_t$, compute the optimal query cluster size $k(v_{t,i})$ maximizing $\mathbf{RSCONF}(\mathbf{NN}(R_t, v_{t,i}, k), \varphi)$, for values of k between a and b
   The ranked collection of patches
   $$\mathcal{C}_t = \langle C_{t,i} \mid i < j \Rightarrow \mathbf{RSCONF}(C_{t,i}, \varphi) \geq \mathbf{RSCONF}(C_{t,j}, \varphi) \rangle$$
   form the candidates for the query clusters associated with sample $R_t \subseteq S$, where $C_{t,i} = \mathbf{NN}(R_t, v_{t,i}, k(v_{t,i}))$ and $C_{t,j} = \mathbf{NN}(R_t, v_{t,j}, k(v_{t,j}))$.
   (c) Let Q_t be a list of patches of $\mathcal{C}_t$ that have been confirmed as query clusters of R_t. Initially, Q_t is empty.
   (d) For all $1 \leq i \leq |\mathcal{C}_t|$ do:
      i. If $\mathbf{RSCONF}(C_{t,i}, \varphi) < \alpha$, then break from the loop.
      ii. For all $w \in C_{t,i}$ do: if $\mathbf{NN}(R_t, w, k) \notin |\mathcal{C}_t|$ for any value of k, or failing that, if $\max\{\mathbf{CONF}(\mathbf{NN}(R_t, w, k), C_{t,i}), \mathbf{CONF}(C_{t,i}, \mathbf{NN}(R_t, w, k))\} < \beta$, then add $C_{t,i}$ to the list Q_t.
3. Let h' be the largest index for which $|Q_{h'}| > 0$. Let $\{C_{t,j}\}$ be the set of patches comprising Q_t, where $C_{t,j} = \mathbf{NN}(R_t, q_{t,j}, k(q_{t,j}))\}$, for all $0 \leq t \leq h'$. Initialize the node set of the query cluster graph G to contain these patches, one patch per node.
4. For all $\delta \leq t \leq h'$, all $1 \leq j \leq |Q_t|$, and all $\max\{0, t-\delta\} \leq s \leq t$, do:
   (a) Compute $C'_{t,j} = \mathbf{NN}(R_s, q_{t,j}, 2^{t-s} k(q_{t,j}))\}$.
   (b) For all $1 \leq i \leq |Q_s|$, if $C_{s,i} \neq C'_{t,j}$ and $\max\{\mathbf{CONF}(C_{s,i}, C'_{t,j}), \mathbf{CONF}(C'_{t,j}, C_{s,i})\} \geq \gamma$, then introduce the edges $(C_{s,i}, C_{t,j})$ and $(C_{t,j}, C_{s,i})$ into G, with weights $\mathbf{CONF}(C_{s,i}, C'_{t,j})$ and $\mathbf{CONF}(C'_{t,j}, C_{s,i})$, respectively.

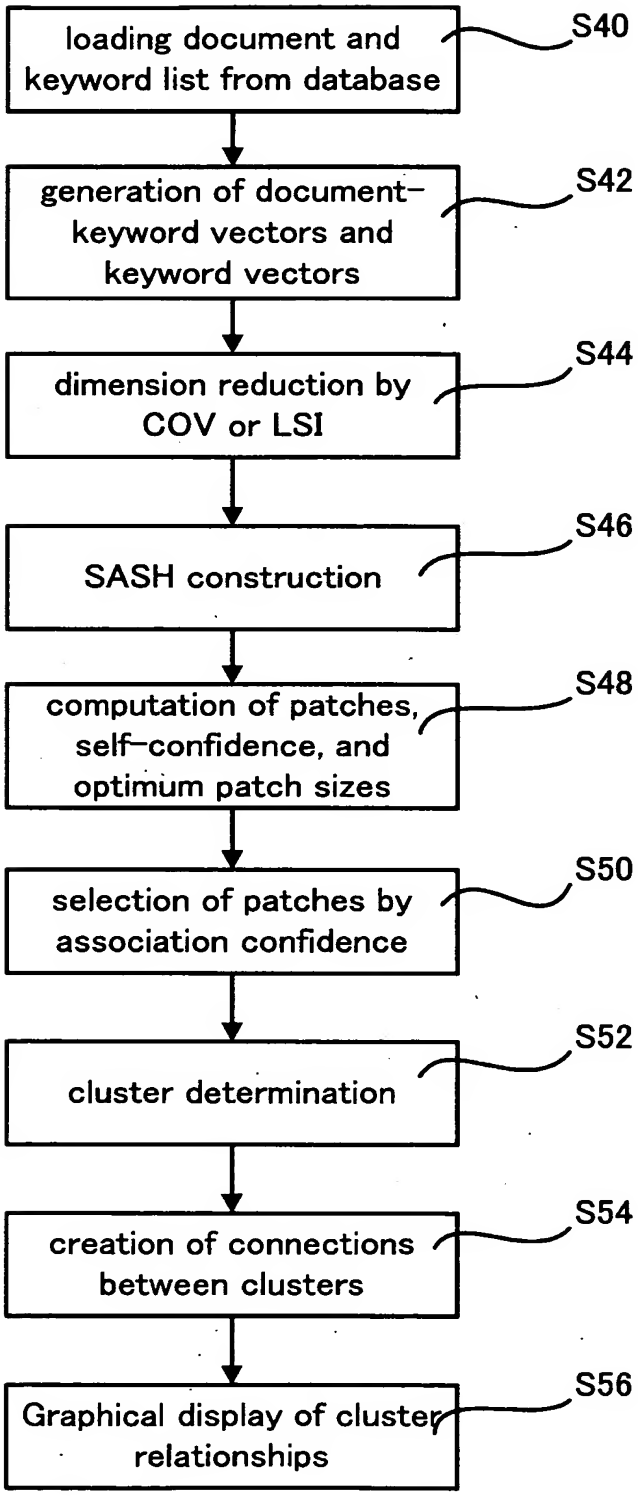
【Fig. 10】



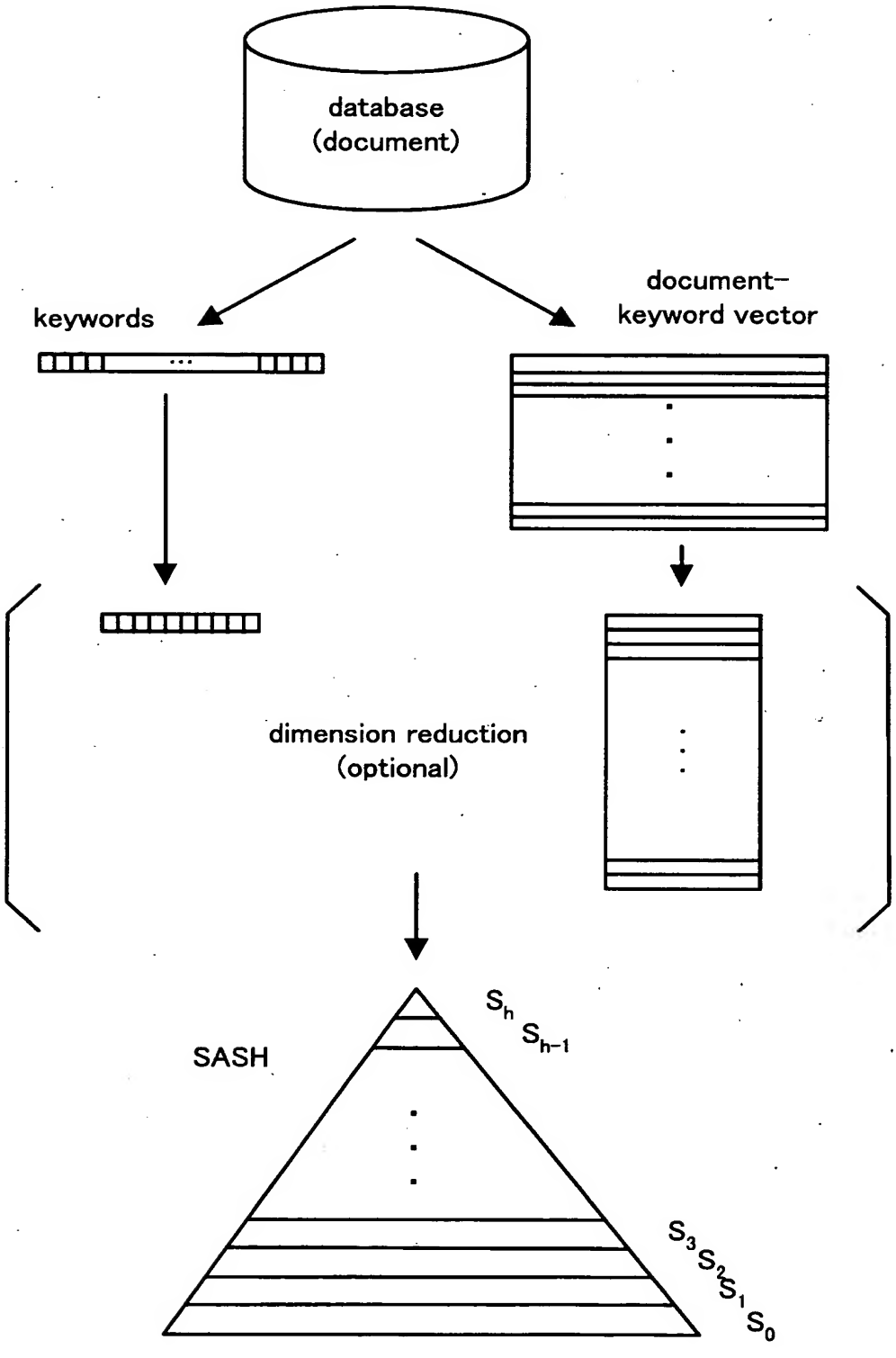
【Fig. 11】


display
GUI data generation
42
cluster estimation
28
44
queried patch generation
30
SASH
retrieval
40
32
hard disk (patch & SCONF list repository)
SASH generation
36
patch definition
confidence determination
38
46
query vector generation
34
Document vector generation
24
26
input query
database (documents)
18

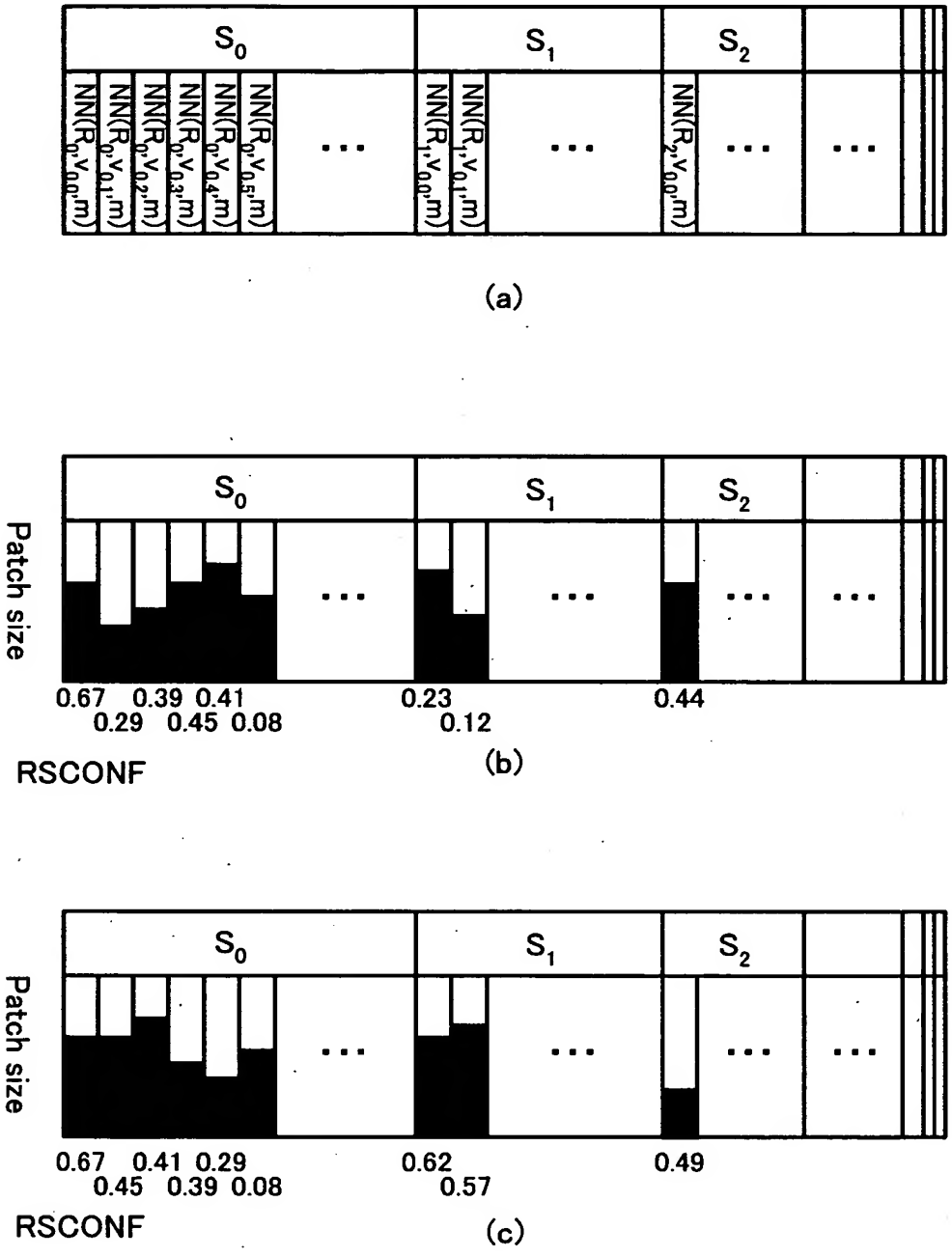
【Fig. 12】


loading document and keyword list from database
S40
generation of document-keyword vectors and keyword vectors
S42
dimension reduction by COV or LSI
S44
SASH construction
S46
computation of patches, self-confidence, and optimum patch sizes
S48
selection of patches by association confidence
S50
cluster determination
S52
creation of connections between clusters
S54
Graphical display of cluster relationships
S56

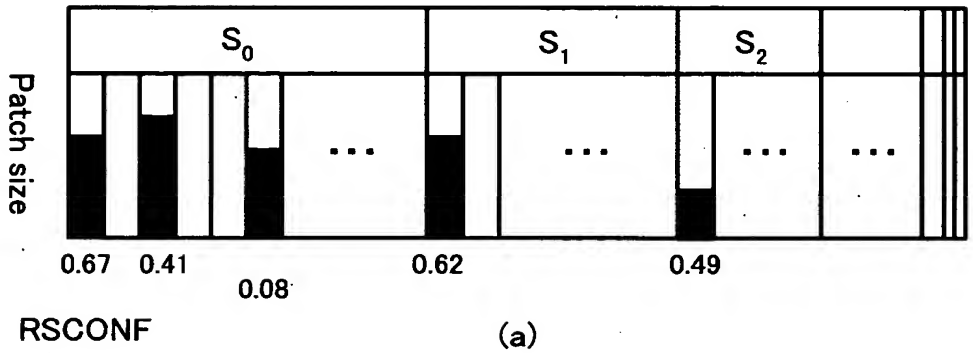
【Fig. 13】

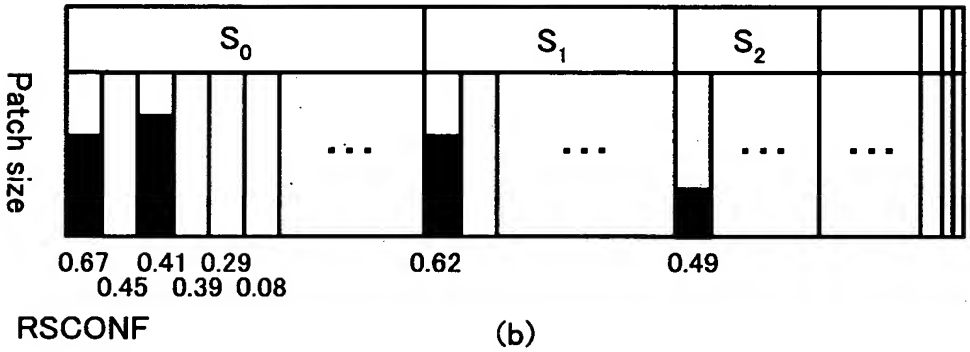

database
(document)
keywords
document-
keyword vector
dimension reduction
(optional)
SASH
S_h S_{h-1}
S_3 S_2 S_1 S_0

【Fig. 14】

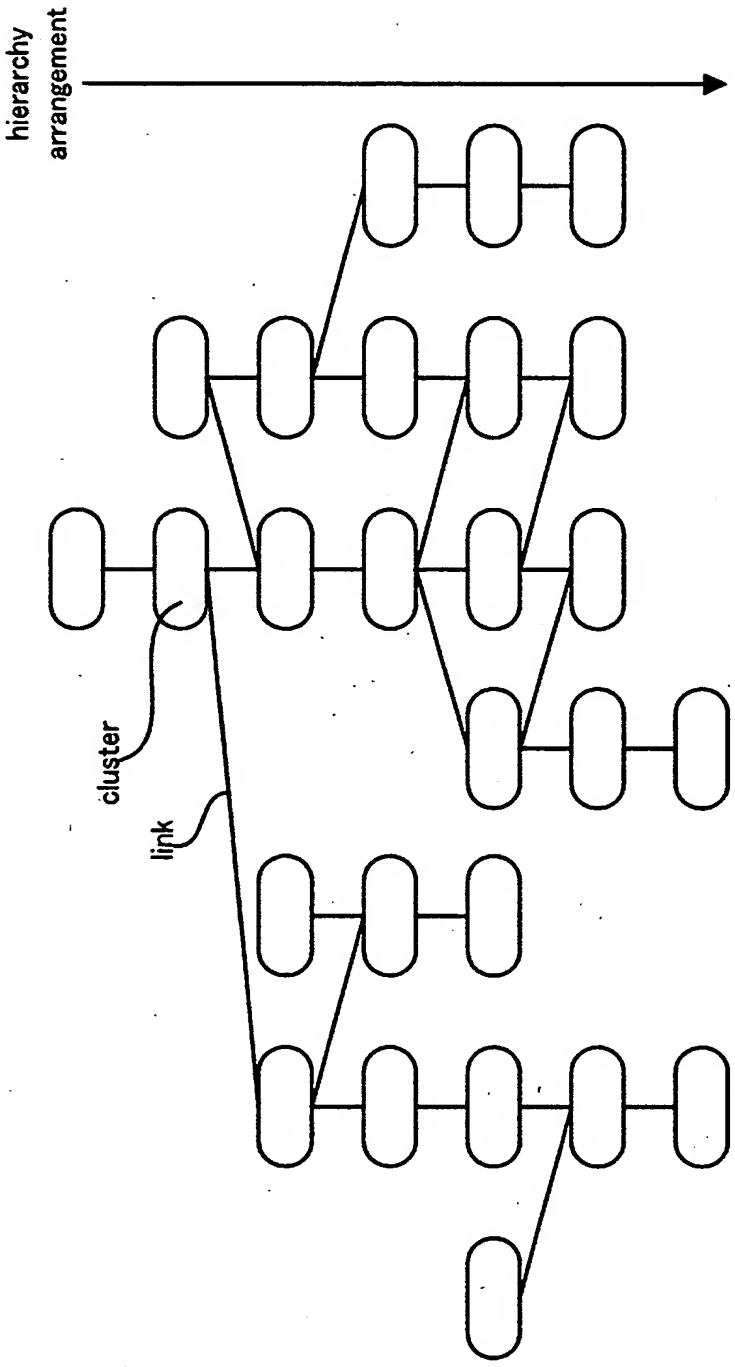

S0
S1
S2
NN(R0,V0,0,m)
NN(R0,V0,1,m)
NN(R0,V0,2,m)
NN(R0,V0,3,m)
NN(R0,V0,4,m)
NN(R0,V0,5,m)
NN(R1,V0,0,m)
NN(R1,V0,1,m)
NN(R2,V0,0,m)
(a)
Patch size
0.67 0.39 0.41
0.29 0.45 0.08
0.23
0.12
0.44
RSCONF
(b)
Patch size
0.67 0.41 0.29
0.45 0.39 0.08
0.62
0.57
0.49
RSCONF
(c)

【Fig. 15】

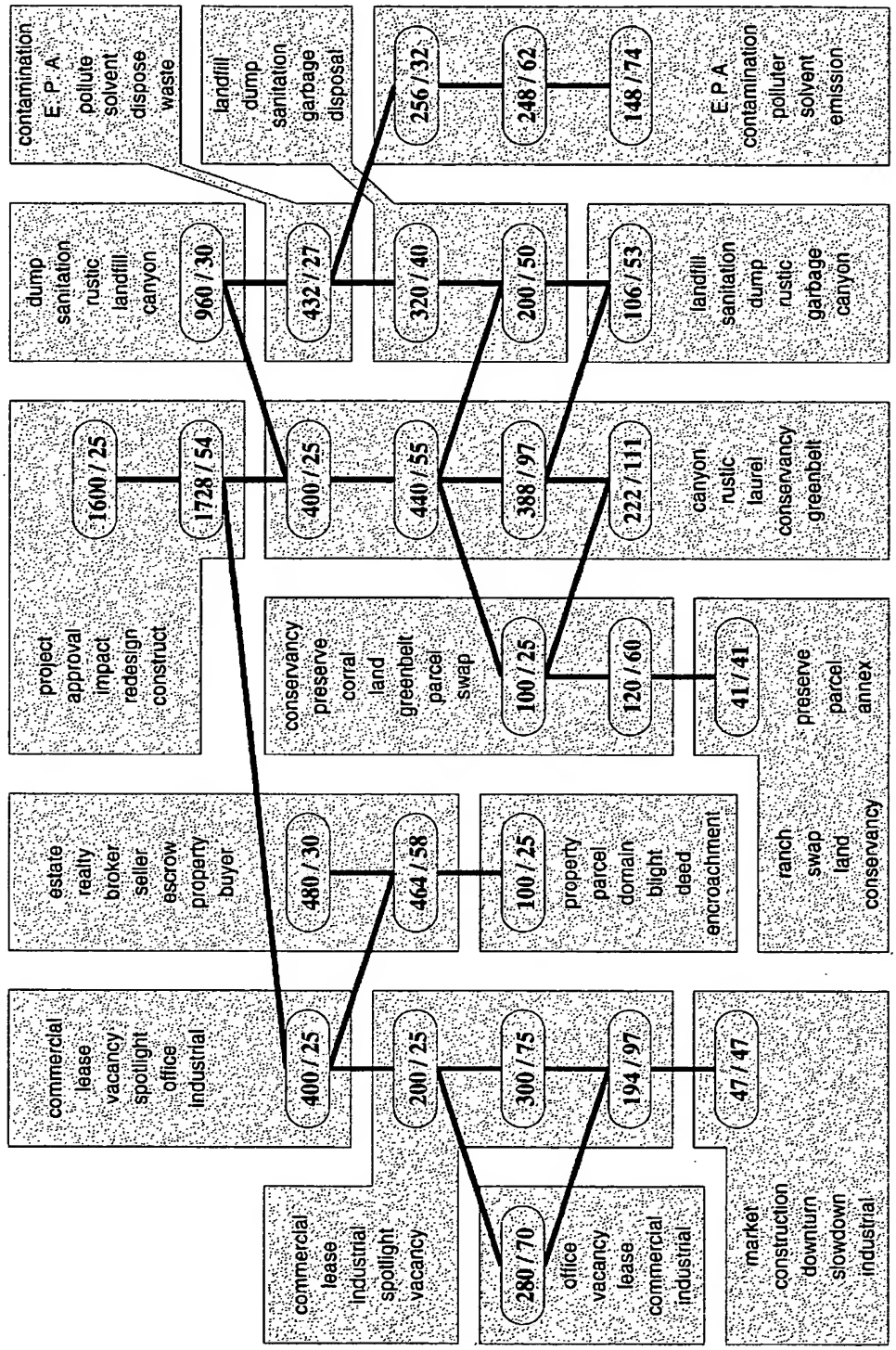

S0
S1
S2
Patch size
0.67
0.41
0.08
0.62
0.49
RSCONF
(a)


S0
S1
S2
Patch size
0.67
0.41
0.29
0.45
0.39
0.08
0.62
0.49
RSCONF
(b)

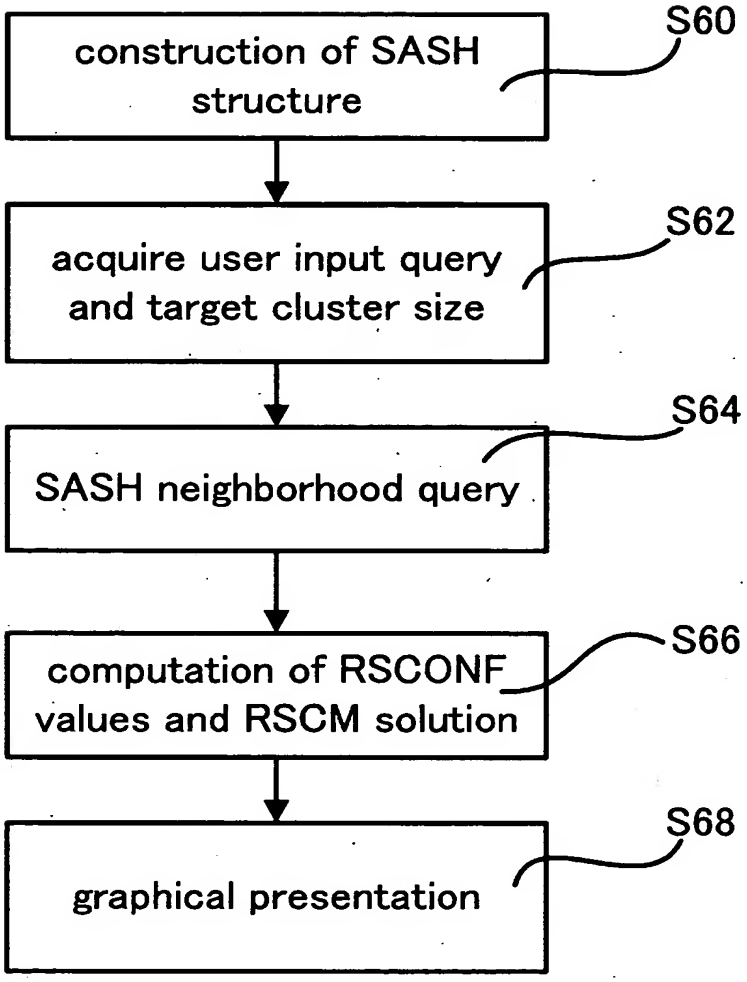
【Fig. 16】



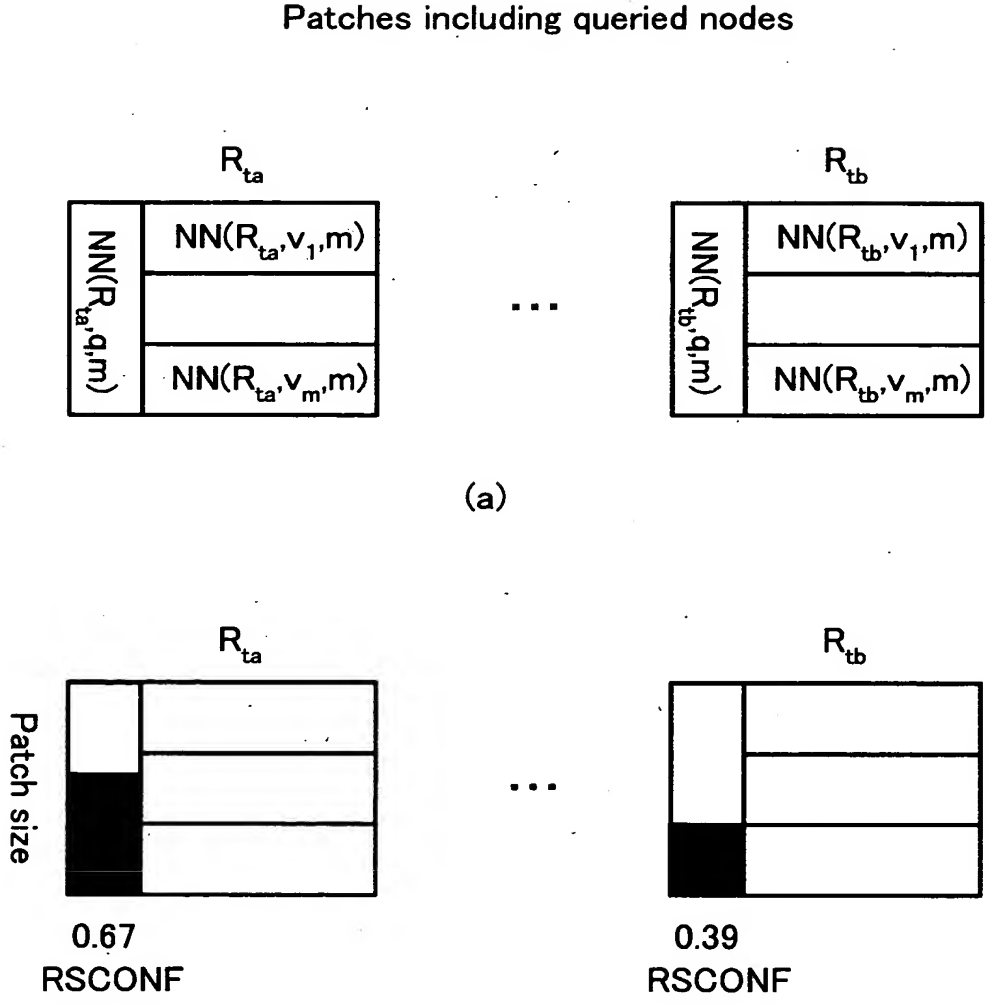
【Fig. 17】


contamination E.P.A. pollute solvent dispose waste
landfill dump sanitation garbage disposal
256 / 32
248 / 62
148 / 74
E.P.A. contamination polluter solvent emission
dump sanitation rustic landfill canyon
960 / 30
432 / 27
320 / 40
200 / 50
106 / 53
landfill sanitation dump rustic garbage canyon
1600 / 25
1728 / 54
400 / 25
440 / 55
388 / 97
222 / 111
canyon rustic laurel conservancy greenbelt
project approval impact redesign construct
conservancy preserve corral land greenbelt parcel swap
100 / 25
120 / 60
41 / 41
preserve parcel annex
estate realty broker seller escrow property buyer
480 / 30
464 / 58
100 / 25
property parcel domain blight deed encroachment
ranch swap land conservancy
commercial lease vacancy spotlight office industrial
400 / 25
200 / 25
300 / 75
194 / 97
47 / 47
commercial lease industrial spotlight vacancy
280 / 70
office vacancy lease commercial industrial
market construction downturn slowdown industrial

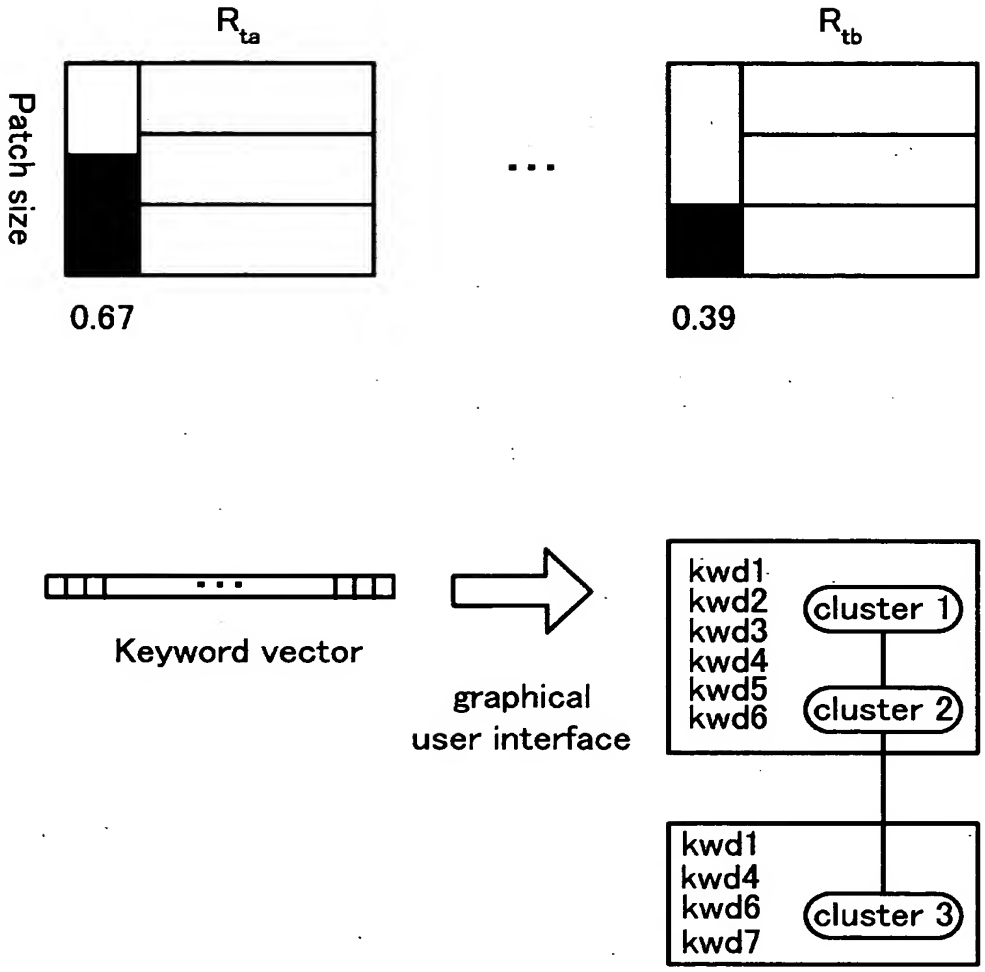
【Fig. 18】


construction of SASH structure
S60
acquire user input query and target cluster size
S62
SASH neighborhood query
S64
computation of RSCONF values and RSCM solution
S66
graphical presentation
S68

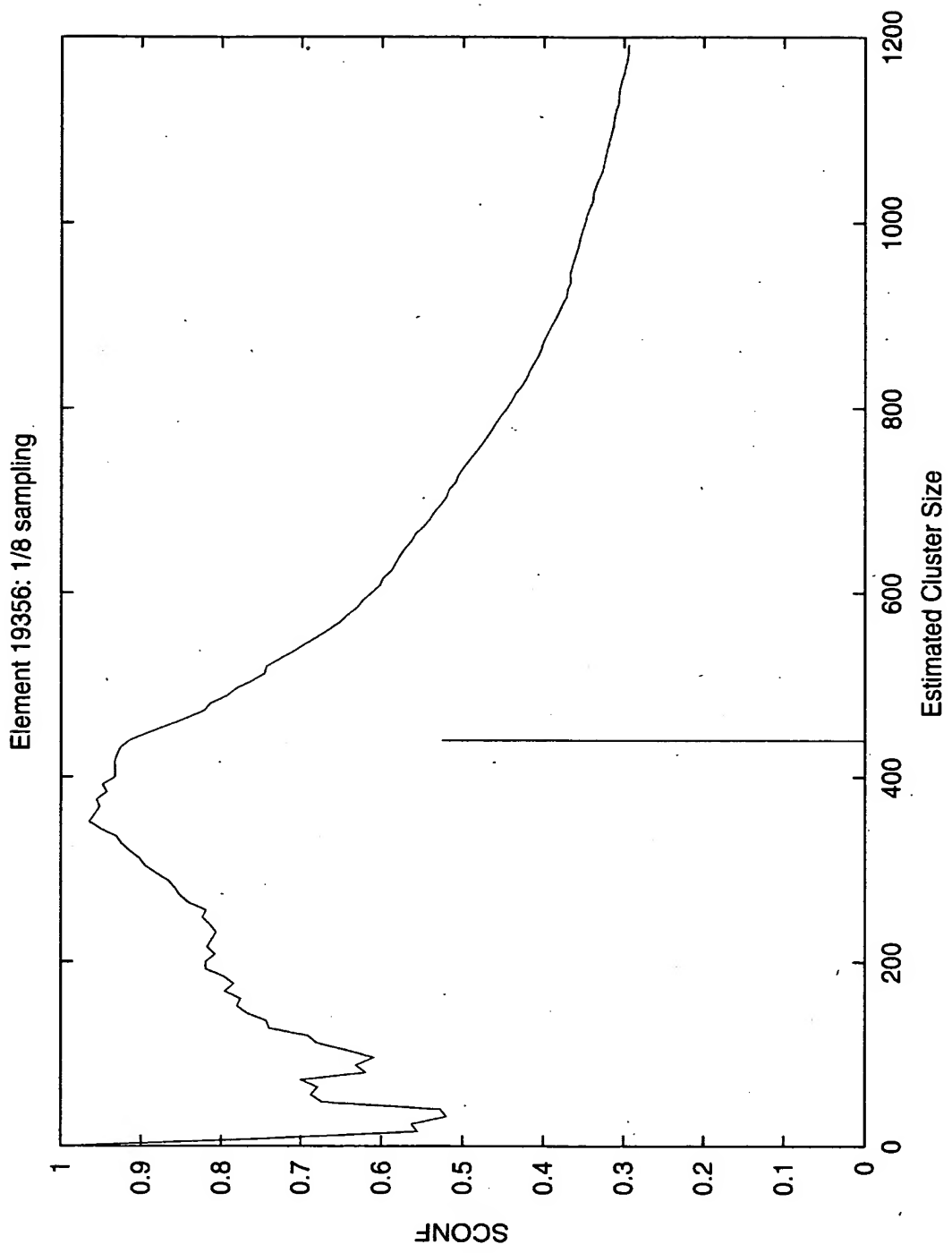
【Fig. 19】


Patches including queried nodes
R_{ta}
R_{tb}
$NN(R_{ta},q,m)$
$NN(R_{ta},v_1,m)$
$NN(R_{ta},v_m,m)$
$NN(R_{tb},q,m)$
$NN(R_{tb},v_1,m)$
$NN(R_{tb},v_m,m)$
(a)
Patch size
0.67
RSCONF
0.39
RSCONF
(b)

【Fig. 20】


R_{ta}
R_{tb}
Patch size
0.67
0.39
Keyword vector
graphical
user interface
kwd1
kwd2
kwd3
kwd4
kwd5
kwd6
cluster 1
cluster 2
kwd1
kwd4
kwd6
kwd7
cluster 3

【Fig. 21】


Element 19356: 1/8 sampling
SCONF
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
200
400
600
800
1000
1200
Estimated Cluster Size

【書類名】　外国語要約書

[Abstract]

[Objective] To provide a computer system for generating data structure for information retrieval, a method thereof, a computer executable program for generating a data structure for information retrieval, a computer readable medium storing the program for generating a data structure for information retrieval, an information retrieval system, and a graphical user interface system.

[Means to Solve Problem] A computer system for generating data structures for information retrieval of documents stored in a database, the computer system comprising: a neighborhood patch generation part 34 for defining patch of nodes having predetermined similarities in a hierarchy structure. The neighborhood patch generation part 34 comprises a hierarchy generation part 36 for generating a hierarchy structure upon the document-keyword vectors and a patch definition part 26. The computer system also comprises a cluster estimation part 28 for generating cluster data of the document-keyword vectors using the similarities of patches.

[Selected Drawings] Fig. 11

認定・付加情報

特許出願の番号	特願２００２－３６８２７６
受付番号	５０２０１９２６７５５
書類名	特許願
担当官	佐々木　吉正　　２４２４
作成日	平成１５年　２月２４日

＜認定情報・付加情報＞

【特許出願人】

【識別番号】　390009531

【住所又は居所】　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　ニュー　オーチャード　ロード

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】　100086243

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内

【氏名又は名称】　坂口　博

【代理人】

【識別番号】　100091568

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内

【氏名又は名称】　市位　嘉宏

【代理人】

【識別番号】　100108501

【住所又は居所】　神奈川県大和市下鶴間１６２３番１４　日本アイ・ビー・エム株式会社　知的所有権

【氏名又は名称】　上野　剛史

【復代理人】　申請人

【識別番号】　100110607

【住所又は居所】　神奈川県大和市中央林間３丁目４番４号　サクライビル４階　間山国際特許事務所

【氏名又は名称】　間山　進也

次頁無

【書類名】　翻訳文提出書
【整理番号】　JP9020208
【提出日】　平成15年 2月19日
【あて先】　特許庁長官殿
【出願の表示】
　【出願番号】　特願2002-368276
【特許出願人】
　【識別番号】　390009531
　【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション
【代理人】
　【識別番号】　100086243
　【弁理士】
　【氏名又は名称】　坂口　博
【代理人】
　【識別番号】　100091568
　【弁理士】
　【氏名又は名称】　市位　嘉宏
【代理人】
　【識別番号】　100108501
　【弁理士】
　【氏名又は名称】　上野　剛史
【復代理人】
　【識別番号】　100110607
　【弁理士】
　【氏名又は名称】　間山　進也
【確認事項】　本書に添付した翻訳文は、外国語書面出願の願書に添付した外国語明細書、外国語図面及び外国語要約書に記載した事項を過不足なく適正な日本語に翻訳したものであ

る。

【提出物件の目録】

【物件名】　外国語明細書の翻訳文　1

【物件名】　外国語図面の翻訳文　1

【物件名】　外国語要約書の翻訳文　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　情報検索のためのデータ構造を生成するコンピュータ・システム、そのための方法、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラム、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラムを記憶したコンピュータ可読な記憶媒体、情報検索システム、およびグラフィカル・ユーザ・インタフェイス・システム

【特許請求の範囲】

【請求項１】　所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメントーキーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成するコンピュータ・システムであって、前記コンピュータ・システムは、

前記ドキュメントーキーワード・ベクトルに対して階層構造を生成させる部分と前記ノード間にノード間の距離尺度に関連してパッチの関係を生成するパッチ規定部とを含んで構成され、階層構造を使用して決定される類似性を有するノード・グループを生成するための近傍パッチ生成部と、

前記パッチの間の類似性を使用して前記ドキュメントーキーワード・ベクトルのクラスター・データを生成するためのクラスター見積もり部とを含む

コンピュータ・システム。

【請求項２】　前記コンピュータ・システムは、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を決定するコンフィデンス決定部を含み、前記クラスター見積もり部は、前記パッチ間コンフィデンス値に基づいて前記パッチを選択して前記ドキュメントーキーワード・ベクトルのクラスターとする、請求項１に記載のコンピュータ・システム。

【請求項３】　前記クラスター見積もり部は、前記パッチ内コンフィデンス値に応じて前記クラスターのサイズを見積もる、請求項１に記載のコンピュータ・システム。

【請求項４】　所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメントーキーワード・ベク

トルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成する方法であって、前記方法は、

前記ドキュメントーキーワード・ベクトルに対して階層構造を生成し、格納領域に階層データを格納するステップと、

前記階層構造を使用して決定された類似性を有するノードの近傍パッチを生成し、前記パッチを格納領域に格納するステップと、

前記階層データおよび前記パッチとを読み込んで、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を算出し、前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値を格納領域に対応するリストとして格納するステップと、

前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値に応答して前記パッチを選択し、前記ドキュメントーキーワード・ベクトルのクラスターとするステップと

を含む方法。

【請求項５】　さらに、前記パッチ内コンフィデンス値に応じて前記クラスターのサイズを見積もるステップを含む、請求項４に記載の方法。

【請求項６】　所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメントーキーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成するための方法をコンピュータ・システムに実行させるためのプログラムであって、前記プログラムは、前記コンピュータ・システムに対して

前記ドキュメントーキーワード・ベクトルに対して階層構造を生成し、格納領域に階層データを格納するステップと、

前記階層構造を使用して決定された類似性を有するノードの近傍パッチを生成し、前記パッチを格納領域に格納するステップと、

前記階層データおよび前記パッチとを読み込んで、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を算出し、前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値を格納領域に対応するリストとして格納するステップと、

前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値に応答して前記パッチを選択し、前記ドキュメントーキーワード・ベクトルのクラスターとするステップと

を実行させるプログラム。

【請求項７】　さらに、前記パッチ間コンフィデンス値に応じて前記クラスターのサイズを見積もるステップを実行させる、請求項６に記載のプログラム。

【請求項８】　所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメントーキーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成するための方法をコンピュータ・システムに実行させるプログラムが記録されたコンピュータ可読な媒体であって、前記プログラムは、前記コンピュータ・システムに対して

前記ドキュメントーキーワード・ベクトルに対して階層構造を生成し、格納領域に階層データを格納するステップと、

前記階層構造を使用して決定された類似性を有するノードの近傍パッチを生成し、前記パッチを格納領域に格納するステップと、

前記階層データおよび前記パッチとを読み込んで、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を算出し、前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値を格納領域に対応するリストとして格納するステップと、

前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値に応答して前記パッチを選択し、前記ドキュメントーキーワード・ベクトルのクラスターとするステップと

を実行させるコンピュータ可読な記憶媒体。

【請求項９】　さらに、前記パッチ間コンフィデンス値に応じて前記クラスターのサイズを見積もるステップを実行させる、請求項８に記載の記憶媒体。

【請求項１０】　所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメントーキーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のための情報検

索システムであって、前記情報検索システムは、

前記ドキュメントーキーワード・ベクトルに対して階層構造を生成させる部分と前記ノード間にノード間の距離尺度に関連してパッチの関係を生成するパッチ規定部とを含んで構成され、構造階層を使用して決定された類似性を有するノード・グループを生成するための近傍パッチ生成部と、

前記パッチの間の類似性を使用して前記ドキュメントーキーワード・ベクトルのクラスター・データを生成するためのクラスター見積もり部と、

前記見積もられたクラスター・データをディスプレイ手段に表示させるグラフィカル・ユーザ・インタフェイスと

を含む情報検索システム。

【請求項１１】　前記情報検索システムは、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を決定するコンフィデンス決定部を含み、前記クラスター見積もり部は、前記パッチ間コンフィデンス値に基づいて前記パッチを選択して前記ドキュメントーキーワード・ベクトルのクラスターとする、請求項１０に記載の情報検索システム。

【請求項１２】　前記クラスター見積もり部は、前記パッチ間コンフィデンス値に応じて前記クラスターのサイズを見積もる、請求項１０に記載の情報検索システム。

【請求項１３】　前記システムは、さらに、クエリーを受信すると共に情報検索のためのデータを抽出してクエリー・ベクトルを生成するユーザ・クエリー受信部と、前記ドキュメントーキーワード・ベクトルと前記クエリー・ベクトルとの間の類似性を算出する情報検索部とを含む、請求項１０に記載の情報検索システム。

【請求項１４】　前記クラスターは、ユーザ入力クエリーに関連して検索された前記ドキュメントーキーワード・ベクトルを使用して見積もられる、請求項１３に記載の情報検索システム。

【請求項１５】　ユーザ入力クエリーに応答してディスプレイ・デバイス上に見積もられたクラスターをグラフィカルに表示するためのグラフィカル・ユーザ・インタフェイス・システムであって、前記グラフィカル・ユーザ・インタフェ

イス・システムは、

ドキュメントを格納するデータベースと、

前記データベースに格納された前記ドキュメントについてドキュメント－キーワード・ベクトルを生成すると共に、前記ユーザ入力クエリーに応答してドキュメントのクラスターを見積もるためのコンピュータと、

前記見積もられたクラスターと共に、前記クラスターの間のコンフィデンス関係とクラスター・サイズの階層情報とを画面上に表示するためのディスプレイとを含むグラフィカル・ユーザ・インタフェイス・システム。

【請求項１６】　前記コンピュータは、

前記ドキュメント－キーワード・ベクトルに対して階層構造を生成させる部分と前記ノード間にノード間の距離尺度に関連してパッチ関係を生成するパッチ規定部とを含み、検索構造を使用して決定される類似性を有するノード・グループを生成するための近傍パッチ生成部と、

前記パッチの間の類似性を使用して前記ドキュメント－キーワード・ベクトルのクラスター・データを生成するためのクラスター見積もり部とを含む、請求項１５に記載のグラフィカル・ユーザ・インタフェイス・システム。

【請求項１７】　前記コンピュータは、

前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を決定するコンフィデンス決定部を含み、前記クラスター見積もり部は、前記パッチ間コンフィデンス値に基づいて前記パッチを選択して前記ドキュメント－キーワード・ベクトルのクラスターとする、請求項１６に記載のグラフィカル・ユーザ・インタフェイス・システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、大規模データベースからの情報検索に関し、より詳細には、情報検索のためのデータ構造を生成するためのコンピュータ・システム、そのための方法、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラム、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラムを

記憶したコンピュータ可読な記憶媒体、情報検索システム、およびグラフィカル・ユーザ・インタフェイス・システムに関する。

【０００２】

【従来の技術】

近年における情報処理システムは、例えば、ニュース・データ、顧客情報、特許情報および株式市場データと言った大量のデータを取り扱うことがますます期待されている。上述したデータベースのユーザは、所望するデータの迅速、効果的、かつ高精度な検索を行うことがますます困難になってきている。このため、大規模データベースからの適時的で精度の良いドキュメントの安価な検出を行うことは、多くの種類の事業に対してきわめて価値ある情報を提供することになろう。加えて、ユーザは、時として検出されたデータに関連する例えば、データベース中におけるクラスター情報およびクラスター間にわたる相関関係などについて、さらなる情報を得ることを希望する場合もある。

【０００３】

クラスターを検出するための典型的な方法は、データ要素間の類似の尺度に依存し、類似検索に基づく上述した方法がこれまで提案されており、これらについては以下に挙げることができる。

【０００４】

類似検索（また、曖昧検索としても知られる）は、データベース内のアイテムが与えられたクエリーに対してどの程度類似するかを探索するものである。類似（または、非類似）は、典型的にはある種の実数値または整数値の距離である“尺度”を使用した距離(dist)、すなわち、

【０００５】

(1) dist(p, q) ≧ 0 for all p, q (non-negativity);

(2) dist(p, q) = dist(q, p) for all p, q (symmetry);

(3) dist(p, q) = 0 if and only if p = q;

(4) dist(p, q) + dist(q, r) ≧ dist(p, r) （すべてのp, q, r に対して）

（三角不等式）

を使用することにより、モデリングされる。

【０００６】

上述した距離関数が存在する対象のセットは、いずれも距離空間として参照することができる。クエリー時において、距離評価の数を低減することが可能なデータ構造は、インデックスとして知られている。類似クエリーのための多くの方法が提案されている。距離空間における類似クエリーとしては、下記のような概ね２通りが知られている。

（Ａ）ｋ－近接クエリー：クエリー要素ｑと正の正数ｋとを与え、ｑに対して近い方からｋ番目のデータベース要素を報告する。

（Ｂ）レンジ・クエリー：クエリー要素ｑと、距離ｒとを与え、$dist(p, q) \leqq r$となるアイテムｑを報告するというものである。

【０００７】

大規模データベースについて言えば、類似クエリーを、クエリー要素からデータベース要素のすべてについて正確に算出することはきわめてコスト的に高いものとなる。データベース要素間のすべての距離を算出し、格納する従来の方法は、データベース要素の数の２乗に比例する時間と記憶空間とを必要とするので、コスト的に高すぎる（すなわち、計算時間とメモリ使用量とに対して２乗のオーダーとなる）。より現実的には、２次以下の記憶空間および前処理時間を使用して時間に対して略直線的なクエリー処理を可能とする検索構造を構築することが目的となりうる。

【０００８】

Ａ．ベクトル空間モデルの概説

近年における情報検索方法は、多くの場合、ベクトル空間モデルを使用してデータベースのドキュメントを表示させる。このようなベクトル空間モデルにおいては、考察しているデータベース内の各ドキュメントはベクトルを伴い、ベクトルの各座標軸は、ドキュメントのキーワード、すなわちアトリビュートを示す。ベクトル空間モデルの詳細については、別の文献に与えられている（Gerald Salton, The SMART Retrieval System - Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, NJ, USA, 1971）。

【０００９】

Ｂ．類似検索構造の概説

過去３０年にわたり、類似クエリーを取り扱うための多くの種類の構造が提案されてきている。これらのうちの主要なものは、空間インデックスであり、このインデックスは、d個の実数値のアトリビュート・ベクトルとして対象セットがモデルされていることを必要とするものである。他のものは、“距離”インデックスであり、距離インデックスは、距離尺度の存在以外にはデータベース要素の性質に対してまったく仮定をすることがなく、したがって空間検索構造よりもより広い適用性がある。多次元ベクトル空間および距離空間のための検索構造についての最近の総説については、Gaedeら (Volker Gaede and Oliver Gunther, Multidimensional Access Methods, ACM Computing Surveys, 30, 2, 1998, pp. 170-231.), and Chavez ら (Edgar Chavez, Gonzalo Navarro, Ricardo Baeza-Yates and Jose L. Marroquin, Searching in metric spaces, ACM Computing Surveys 33, 3, 2001, pp. 273-321.)を参照されたい。

【００１０】

類似検索は、距離データまたはベクトル・データであるにせよ、“次元の呪い”として参照される効果により、多くの場合に制限を受ける。近年における最近傍クエリーまたは範囲クエリーの計算の一般的な問題は、低いフラクタル分布、本質的な次元の低さ、または分布の他の特性など、空間的な特性を有するデータの分布に基づかない限り、正確な技術でさえもがデータベース全体の連続的な検索に対して実質的な改善をもたらすものではない、ということが証明されている。

【００１１】

データの次元および次元の呪いについてのさらなる情報については、例えば、Chavezら （前掲）, Pagelら (Bernd-Uwe Pagel, Flip Korn and Christos Faloutsos, Deflating the dimensionality curse using multiple fractal dimensions, Proc. 16th International Conference on Data Engineering (ICDE 2000), San Diego, USA, IEEE CS Press, 2000, pp. 589-598.), Pestov (Vladimir Pestov, On the geometry of similarity search: dimensionality curse and concentration of measure, Information Processing Letters, 73, 2000, pp. 47

-51.), and Weberら (Roger Weber, Hans-J. Schek and Stephen Blott, A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, Proc. 24th VLDB Conference, New York, USA, 1998, pp. 194-205)を参照されたい。

【００１２】

Ｃ．近似的な類似検索の概説

次元の呪いを回避する試みとして、研究者は、計算の高速化を得ることを希望し、類似クエリーを幾分か犠牲にすることを考案した。これらの技術の詳細については他の文献、例えばIndykら (P. Indyk and R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, Proc. 30th ACM Symposium on Theory of Computing, Dallas, 1998, pp. 604-613.), and Ferhatosmanogluら (Hakan Ferhatosmanoglu, Ertem Tuncel, Divyakant Agrawal and Amr El Abbadi, Approximate nearest neighbor searching in multimedia databases, Proc. 17th International Conference on Data Engineering (ICDE), Heidelberg, Germany, IEEE CS Press, 2001, pp. 503-514.); 距離空間については、Ciacciaら (Paolo Ciaccia and Marco Patella, PAC nearest neighbor queries: approximate and controlled search in high-dimensional and metric spaces, Proc. 16th International Conference on Data Engineering (ICDE 2000), San Diego, USA, 2000, pp. 244-255; Paolo Ciaccia, Marco Patella and Pavel Zezula, M-tree: an efficient access method for similarity search in metric spaces, Proc. 23rd VLDB Conference, Athens, Greece, 1997, pp. 426-435.) and Zezulaら (Pavel Zezula, Pasquale Savino, Giuseppe Amato and Fausto Rabitti, Approximate similarity retrieval with M-trees, The VLDB Journal, 7, 1998, pp. 275-293.)を参照されたい。しかしながら、これらの方法は、すべて実際上の有効性に制限を与えるという不都合をもたらすものである。あるものは、データの分布について非現実的な仮定を行ない、他のものは、精度と速度との間のトレード・オフに対して効果的な管理を行うことができないものである。

【００１３】

Ｄ．空間近似サンプル階層(SASH)

Houleら(Michael E. Houle, SASH: a spatial approximation sample hierarchy for similarity search, IBM Tokyo Research Laboratory Research Report RT-0446, 18 pages, February 18, 2002)およびHoule, KobayashiおよびAono (Japanese Patent Application No. 2002-037842)による大規模な多次元データ・セットのための近似的な類似検索構造は、精度－速度トレード・オフに対して著しく良好な制御を可能とするものであり、これを空間近似サンプル階層(SASH)として参照する。SASHは、距離尺度の条件を満足する類似関数を必要とするものの、データの性質に対してはそれ以外の仮定を加えない。各データ要素には、構造内における固有の位置が与えられ、２つの要素間のそれぞれの連結は、それらが密接な関係にあることを示す。階層構造の各レベルは、要素のランダム・サンプルとして構成され、また各レベルのサンプル・サイズは、直上のレベルの概ね２倍となるように構成されている。この構造は、所与の要素vに最も近い複数の要素をvに対して最も類似するものとして組織化される。具体的には、vに対応するノードは、その上のレベルの近傍点のセットへと結合され、またvが近傍点として選択される下層のアイテムのセットへと関連づけられる。

【００１４】

Ｅ．クラスター化技術の概説

用語「クラスター化」は、類似基準にしたがい、ラベル付けされていないデータのいずれかのグループ分けを意味する。従来のクラスター化方法は、概ね分割または階層化に分類することができる。階層化技術は、データのグループ（クラスター）の包含関係を示す木構造を生成し、木の経路がデータ・セット全体に対応する。分割技術は、典型的には関連性のないクラスターの、固定された数の中からデータ点の分布における分類誤差を大局的に最小にすることに基づくものである。これらの最近の概説（Jain, Murty and Flynn （A. K. Jain, M. N. Murty and P. J. Flynn, Data clustering: a review, ACM Computing Surveys 31, 3, 1999, pp. 264-323.））においては、分割クラスター化手法は、階層化よりもコスト的に高くはならないものの、また柔軟性が大きく劣るとしている。簡略化され、迅速性（時間と複雑さとの間の線形関係が見出される。）と、実装化の

容易性とはあるものの、周知の分割アルゴリズムであるK-meansとその変形法でさえも、概ね大規模なデータ・セットに対して良好に機能しない。分割アルゴリズムは、等質的な（近似された）クラスターの生成には向くものの、不規則な形質のクラスターを見つけるには充分に適するとは言えない。

【００１５】

Ｆ．階層凝集クラスター化

階層凝集クラスター化では、各データ点をまず考察し、分離されたクラスターを構築する。クラスターの対は、その後、すべてのデータ点が１つのクラスターに帰属されるまで連続的に統合される。各ステップで生成されたより大きなクラスターは、統合されたd個のサブクラスターの要素を含んでおり、そしてクラスター階層を与える包含関係とされる。統合する対の選択は、クラスター間距離の所定の基準を最小化させるようにして行われる。

【００１６】

Ｇ．近傍共有方法

単純な距離に基づく凝集クラスタリング方法の基準の１つは、より高い密度のクラスターを形成するように方向付けを行うものである。領域内のデータの良好に随伴する低密度のグループは、多くの対の距離が統合しきい値よりも低い場合には、まったく発見されないというリスクを負う。凝集クラスター化のために、より専用化された（より高コストの）、データ要素間の近接性を考慮した距離尺度が提案されている。Jarvisらは、R. A. Jarvis and E. A. Patrick, Clustering using a similarity measure based on shared nearest neighbors, IEEE Transactions on Computers C-22, 11, Nov. 1973, pp. 1025-1034.において、任意の類似基準distと、固定した整数のパラメータk＞r＞0とで統合基準を規定しており、最近傍点の少なくとも所定数が同一のクラスターにより共有される場合には、２つのデータ要素は、それら自体が同一のクラスターであるとされる。クラスターを統合するか否か、の決定は、データ・セットの局所的な密度には依存せず、実質的な仕方で近傍を共有する互いの要素の対が存在するか否かに依存することとなる。

【００１７】

JarvisとPatrickの方法（前掲）は、凝集的なものであり、また随伴の連鎖を介した不適正なクラスターを生成してしまう単一リンク法(single-link method)に類似する。より最近に変形例が提案されており、生成されるクラスターの質を変更する試みがなされている。例えば、Guhaら(S. Guha, R. Rastogi and K. Shim, ROCK: a robust clustering algorithm for categorical attributes, Information Systems 25, 5, 2000, pp. 345-366.); by Ertozら(Levent Ertoz, Michael Steinbach and Vipin Kumar, Finding topics in collections of documents: a shared nearest neighbor approach, University of Minnesota Army HPC Research Center Preprint 2001-040, 8 pages, 2001.); Ertozら(Levent Ertoz, Michael Steinbach and Vipin Kumar, A new shared nearest neighbor clustering algorithm and its applications, Proc. Workshop on Clustering High Dimensional Data and its Applications (in conjunction with 2nd SIAM International Conference on Data Mining), Arlington, VA, USA, 2002, pp. 105-115.); Daylight Chemical Information Systems Inc., URLアドレス(http://www.daylight.com/);およびBarnard Chemical Information Ltd., URLアドレス(http://www.bci.gb.com/)を参照されたい。それにもかかわらず、すべての変形例は、依然として凝集アルゴリズムの主要な特徴を示し、随伴性の少ない要素の連結鎖を伴う大きな不適切な形のクラスターの生成を許してしまうものである。

【００１８】

H．次元削減方法の概説

潜在的意味インデキシング(LSI)は、ドキュメントのランク付け問題の次元を削減するためのベクトル空間に基づくアルゴリズムである。これについては、Deerwesterら(Scott Deerwester, Susan T. Dumais, George W. Furnas, Richard Harshman, Thomas K. Landauer, Karen E. Lochbaum, Lynn A. Streeter, Computer information retrieval using latent semantic analysis, U.S. Patent No. 4839853, filed Sept. 15, 1988, issued June 13, 1989; Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science, 41, 6, 1990, pp. 391-407.)を参照されたい。LSIは

、検索およびランク付けの問題を、著しく低次元の問題へと低下させることで、きわめて大規模なデータベースの検索をより効率的に実行させることができる。他の次元削減手法は、Kobayashiら（Mei Kobayashi, Loic Malassis, Hikaru Samukawa, Retrieval and ranking of documents from a database, IBM Japan, docket No. JP9-2000-0075, filed June 12, 2000; Loic Malassis, Mei Kobayashi, Statistical methods for search engines, IBM Tokyo Research Laboratory Research Report RT-413, 33 pages, May 2, 2001.）によりCOVとして参照される次元削減方法として提案されている。COVでは、ドキュメント・ベクトルの共分散行列を使用して、ドキュメント・ベクトルの射影を行う近似的な削減次元空間を決定している。LSIおよびCOVは、情報検索において互換的な方法であり、いくつかのデータベース、およびいくつかのクエリーについてLSIは、わずかに良好な結果をCOVに比較して与えるものの、それ以外についてCOVは、わずかながら良好な結果を与える。

【発明が解決しようとする課題】

さらに、従来の距離に基づくクラスター検出には、後述する他の不都合があることも知られている。

【００１９】

機械が学習するコンテキストの通常のクラスター化法は、データ・セットにおける主要なグループ化を見出すように設計されている。このためこの方法は、クラスターが高精度で未知のポイントを分類することができるのであれば良好なものと考えることができる。しかしながら、データ・マイニングの分野においては、データの主要クラスターは、多くの場合、ユーザには知られてはおらず、価値ある情報を抽出することができる可能性があるのは、より小さく、マイナーなクラスターである、といえる。分割および凝集に基づく既存のクラスター化技術は、多くの場合にバックグラウンドから小さなデータ・クラスターを分離するには効果的ではない。

【００２０】

さらに別の不都合は、大規模なテキスト・データベースは、典型的には情報検索操作を効率化するために小さな集団に分割されていることにある。この分布は

、通常ではデータ・セットの最も大きなクラスターが単一のデータベース内に完全に残されるようにして実行される。しかしながら、主要クラスターに焦点を当てた分割方法は、価値の高いマイナー・クラスターを、いくつかのデータベースの間に分散させてしまうことになる。主要クラスターの他、マイナー・クラスターを同定することは、より効率的にマイナー・クラスターを保存した分割を可能とすることになる。

【００２１】

また、上述したように、クラスター化手段の幾分かのユーザは、しばしばその手段により生成されたクラスターの間の関係を知ることについて興味を有する。クラスターの重なり合いの蓄積を行ない、頂上にデータ・セット全部を含む単一のクラスターがあり、下に向かってより小さなクラスターを配置することにより、階層化クラスター化・アルゴリズムは、上述の要求を満たそうとしている。しかしながら、これらのクラスターのうちの多くは、階層的な組織化の複製物としてのみ存在し、そしてそれら自体では有益な解釈を有するものではない。ユーザは、データ・マイニング手段により報告された各クラスターに対して、ある種の独立した概念的解釈を与えることをまず希望することもある。一旦、意味を有するクラスターのセットが同定されると、ユーザはクラスター間のいずれかの重なり合いや包含関係を知ることに興味を抱くものと考えられる。

【００２２】

加えて、多次元の設定を与えると、ユーザがクラスターについて容易に理解できるような仕方でデータのクラスター随伴性を表示したり、記述したりすることがきわめて困難である。クラスター・データを閲覧する場合に、ユーザは、クラスターの凝集性や突出性の度合いに対し、一目でアクセスすることができる必要性がある。

【００２３】

検索のハードウエア資源に関しても、クラスター化が概ね望ましいことではあるものの、データ・セットがきわめて大規模な場合には高品質のクラスターを得ることに伴う計算コストの故に、データ・マイニング用途のためには現実的ではないものと考えられてきた。通常のコンピュータにおいて有意義な時間内に大規

模なデータ・セットの組織化へのいくらかの知見を提供することができる手段が強く高く必要とされてきた。

【００２４】

上述したように、多くの方法がこれまで提案されているにもかかわらず、高い効率、高い速度と共に高いスケーラビリティを有する情報検索のために好適な新規なデータ構造は、当技術分野で引き続き必要とされてきた。

【００２５】

【課題を解決するための手段】

本発明は、以下において、大規模なテキスト・データベースの情報検索及びデータ・マイニングのための、バックグラウンドに対して高い相互類似の度合いを示す要素（例えばドキュメント）のクラスターの同定に基づいたシステムおよび方法を提案するものである。

【００２６】

本発明においては、クラスターの分布構造は、グラフィカルに表示され、クラスターの質および顕著性として、視覚的に直ちにユーザに対してフィードバックを与えることができる。このシステムは、サイズおよび質といったクラスターの属性に対して自動的にアクセスを可能とする。また、このシステムはユーザに対し、大域的なクラスタリングのための予備的計算を必要とすることなく、クラスターのためのデータ・セットへのクエリーを可能とする。スケーラビリティは、次元削減技術手段、ランダム・サンプリング、および近似的な類似検索を支持するデータ構造の使用により達成される。

【００２７】

本発明は、マイナー・クラスターを保存させつつ、マイナー・クラスターの検出効率を改善することにより、上述した新規な情報検索特性を提供することを可能とする。本発明による新規な情報検索は、ユーザがクラスターを理解する手助けとなるグラフ構造としてクラスター間の相関関係を表現することを可能とする。本発明は、さらに、情報検索の計算の計算スケーラビリティを改善することを可能とするものである。

【００２８】

上述した側面は、クエリー・クラスターを決定するために近傍共有情報を使用する、テキスト・データベースの情報検索およびデータ・マイニングのためのシステムおよび方法により提供される。クラスター化方法は、クエリー要素（データ・セットの要素であっても要素でなくとも良い）と、データ・セットにおけるその近傍との間の相互随伴性のレベルとを評価する。２つの要素の間の随伴性は、これらの要素が共有する最近傍点の比が大きい場合に、強いものとして考察される。従来の共有近傍点の情報を使用する方法とは対照的に、提案する本発明の方法は、クラスター間の随伴性であるコンフィデンス（CONF）およびクラスター内での随伴性であるセルフーコンフィデンス（SCONF）という新たな特有の概念に基づくものである。

【００２９】

本発明によれば、所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメントーキーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成するコンピュータ・システムであって、前記コンピュータ・システムは、

前記ドキュメントーキーワード・ベクトルに対して階層構造を生成させる部分と前記ノード間にノード間の距離尺度に関連してパッチの関係を生成するパッチ規定部とを含んで構成され、階層構造を使用して決定される類似性を有するノード・グループを生成するための近傍パッチ生成部と、

前記パッチの間の類似性を使用して前記ドキュメントーキーワード・ベクトルのクラスター・データを生成するためのクラスター見積もり部とを含む

コンピュータ・システムが提供される。

【００３０】

本発明における前記コンピュータ・システムは、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を決定するコンフィデンス決定部を含み、前記クラスター見積もり部は、前記パッチ間コンフィデンス値に基づいて前記パッチを選択して前記ドキュメントーキーワード・ベクトルのクラスターとすることができる。

【００３１】

本発明の前記クラスター見積もり部は、前記パッチ内コンフィデンス値に応じて前記クラスターのサイズを見積もることができる。

【００３２】

本発明によれば、所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメントーキーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成する方法であって、前記方法は、

前記ドキュメントーキーワード・ベクトルに対して階層構造を生成し、格納領域に階層データを格納するステップと、

前記階層構造を使用して決定された類似性を有するノードの近傍パッチを生成し、前記パッチを格納領域に格納するステップと、

前記階層データおよび前記パッチとを読み込んで、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を算出し、前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値を格納領域に対応するリストとして格納するステップと、

前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値に応答して前記パッチを選択し、前記ドキュメントーキーワード・ベクトルのクラスターとするステップと

を含む方法が提供される。

【００３３】

本発明によれば、所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメントーキーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成するための方法をコンピュータ・システムに実行させるためのプログラムであって、前記プログラムは、前記コンピュータ・システムに対して

前記ドキュメントーキーワード・ベクトルに対して階層構造を生成し、格納領域に階層データを格納するステップと、

前記階層構造を使用して決定された類似性を有するノードの近傍パッチを生成

し、前記パッチを格納領域に格納するステップと、

前記階層データおよび前記パッチとを読み込んで、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を算出し、前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値を格納領域に対応するリストとして格納するステップと、

前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値に応答して前記パッチを選択し、前記ドキュメントーキーワード・ベクトルのクラスターとするステップと

を実行させるプログラムが提供される。

【００３４】

本発明によれば、所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメントーキーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のためのデータ構造を生成するための方法をコンピュータ・システムに実行させるプログラムが記録されたコンピュータ可読な媒体であって、前記プログラムは、前記コンピュータ・システムに対して

前記ドキュメントーキーワード・ベクトルに対して階層構造を生成し、格納領域に階層データを格納するステップと、

前記階層構造を使用して決定された類似性を有するノードの近傍パッチを生成し、前記パッチを格納領域に格納するステップと、

前記階層データおよび前記パッチとを読み込んで、前記パッチ間のパッチ間コンフィデンス値およびパッチ内コンフィデンス値を算出し、前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値を格納領域に対応するリストとして格納するステップと、

前記パッチ間コンフィデンス値および前記パッチ内コンフィデンス値に応答して前記パッチを選択し、前記ドキュメントーキーワード・ベクトルのクラスターとするステップと

を実行させるコンピュータ可読な記憶媒体が提供される。

【００３５】

本発明によれば、所定のキーワード・リストから生成され、かつドキュメントに対して与えられる階層構造のノードを構成するドキュメントーキーワード・ベクトルとしてデータベースに格納されたドキュメントの情報検索のための情報検索システムであって、前記情報検索システムは、

前記ドキュメントーキーワード・ベクトルに対して階層構造を生成させる部分と前記ノード間にノード間の距離尺度に関連してパッチの関係を生成するパッチ規定部とを含んで構成され、階層構造を使用して決定された類似性を有するノード・グループを生成するための近傍パッチ生成部と、

前記パッチの間の類似性を使用して前記ドキュメントーキーワード・ベクトルのクラスター・データを生成するためのクラスター見積もり部と、

前記見積もられたクラスター・データをディスプレイ手段に表示させるグラフィカル・ユーザ・インタフェイスと

を含む情報検索システムが提供される。

【００３６】

本発明によれば、ユーザ入力クエリーに応答してディスプレイ・デバイス上に見積もられたクラスターをグラフィカルに表示するためのグラフィカル・ユーザ・インタフェイス・システムであって、前記グラフィカル・ユーザ・インタフェイス・システムは、

ドキュメントを格納するデータベースと、

前記データベースに格納された前記ドキュメントについてドキュメントーキーワード・ベクトルを生成すると共に、前記ユーザ入力クエリーに応答してドキュメントのクラスターを見積もるためのコンピュータと、

前記見積もられたクラスターと共に、前記クラスターの間のコンフィデンス関係とクラスター・サイズの階層情報とを画面上に表示するためのディスプレイと

を含むグラフィカル・ユーザ・インタフェイス・システムが提供される。

【００３７】

【発明の実施の形態】

パートＩ．本方法の本質的処理

以下に、本発明をドキュメントの情報検索のコンテキストを使用して説明する

が、本発明はこれに制限されるものではなく、本発明のアルゴリズムは、対となった非類似尺度といった距離尺度（三角不等式といった可能な表現）の特性を満足し、各データ要素がキーワードや、アノテーション目的のための他の情報を含む、いかなる用途に対しても適用することができる。上述した用途の１つの例としては、上述した対となった非類似尺度が存在するマルチメディア・データベース（例えば、テキスト、オーディオ、静止画像、グラフィックス・イメージ、グラフィック・ビデオ、ｇｉｆアニメーションなどを含むデータベース）のためのデータ・マイニング・システムを挙げることができる。

【００３８】

本発明の方法の概略的なフローチャートを図１に示す。本発明は、主にテキストに対する適用を用いて説明するが、当業者によれば本発明の方法が、“より近い”（尺度に関連して）要素の対が、“より遠く離れた”要素の対よりもより類似するというどの２つの要素の間においても距離の算出が可能な明確に規定された尺度を含むようにモデル化されたコンテンツを含む、いかなるデータベースに対しても容易に適用することができることは理解されよう。

【００３９】

本発明の方法は、ステップＳ１０から開始し、データベース内のドキュメントがベクトル空間モデルを使用してベクトルへと変換される。次いで、方法は、ステップＳ１２において、データベースに格納されたデータのSASH類似性検索構造を生成する。次いで、ステップＳ１４においてデータベースのすべての要素に対してSASH構造を使用してデータベースの要素に対して、最も類似する要素のリストからなる近傍パッチを算出する。これらのパッチは、その後適切なメモリ領域に格納される。ステップＳ１６においては、セルフ－コンフィデンス値（以下SCONF値として参照する）のリストを、格納されたすべてのパッチについて算出する。これらのSCONF値は、相対セルフ－コンフィデンス値（以下、RSCONF値）を算出するために使用され、このRSCONF値は、各パッチ（それ自身もまたパッチである）の最良のサブセットのサイズを決定するために使用され、クラスター候補が生成される。次いで、本発明の方法は、ステップＳ１８へと進んで、コンフィデンス値（以下、CONF値として参照する。）を使用して冗長なクラスター候補を

排除する。方法はその後ステップＳ２０へと進み、最終的なクラスターとされる所望する最小のRSCONF値を有するクラスター候補の選択をさらに行ない、選択されたクラスターを適切なメモリに格納する。方法はさらにステップＳ２２へと進み、ユーザに対してGUIインタフェイスによりコンピュータ・スクリーン上にクラスターの間の相互関係を示したグラフを表示させる。図１に示した方法は、さらに図１に示した各ステップを実行するための複数のサブステップを含んでおり、これらのサブステップについては、以下においてより詳細に説明する。

【００４０】

＜ドキュメント－キーワード・ベクトルの算出＞

ドキュメント－キーワード・ベクトルは、所与のキーワードおよびドキュメントとから、いくつか知られたうちのいかなる技術を使用しても算出することができる。本発明の特定の実施の形態においては、適切な重み付けを使用してドキュメントを数値化させることもでき、数値化の詳細については他の文献（例えば、Saltonら、前掲）に与えられているので、本発明においては説明を行わない。

【００４１】

＜SASHの構成と使用＞

図２は、空間近似サンプル階層、すなわちSASHとして知られているドキュメント－キーワード・ベクトルの階層構造の構築の手法を示した図である。処理は、図１のステップＳ１０の結果を受け取った後、ステップＳ２８から開始し、例えば周知の乱数発生プログラムを使用してSASHのノードとしてベクトルのランダムな割り当てを生成する。レベルには、0～ｈの数字が割り当てられており、各レベルは、その上のレベルの概ね２倍のベクトル・ノードを含む構成とされている。0の数字が付されたレベルは、データ・セットのベクトル・ノードの概ね半数を含んでおり、hの数字が付されたレベルは、トップ・ノードとして参照される単一のノードを含んでいる。SASH構造のトップ・ノードは、コンピュータ・システムのいずれかに含まれる、いかなる乱数発生手段を使用しても決定されている。次いで、ステップＳ３０においてLで参照される階層レベルを、hに初期化する。プロセスは、ステップＳ３２に進み、階層レベルLを1だけ減少させ、ステップＳ３４においてレベルLのノードを、レベルL+1のノードへと、ノード間の距離に依

存して連結させる。上述した連結は、レベルL+1のノードが親ノードとなり、レベルLのノードが子ノードとなる。連結は、レベルLのノードからレベルL+1からの最も近い親ノードを選択することにより実行され、その後、これらの親－子ノード対を連結して、それぞれの親ノードが所定数の最も近接する子ノードに連結されるようにして行われる。連結をどのようにして行うかのさらに詳細については、Houleらによる他の文献（前掲）を参照されたい。プロセスは、ステップＳ３６へと進み、階層レベルが最も低いレベル(0)に達したか否かを判断し、レベル0の場合(yes)、SASHの構築を完了させ、SASH構造を、メモリまたはハードディスクといった適切な記憶領域へと格納する。プロセスは、ステップＳ３８へと続いてノードのパッチを構築させる。レベル(0)ではない場合(no)には、プロセスは、ステップＳ３２へと戻って、ステップＳ３６において肯定的な結果が得られるまで繰り返される。

【００４２】

ステップＳ３８では、格納されたSASH構造が本発明にしたがって用いられ、データベースのすべての要素についてのパッチが生成される。データベースのサブセットRに関連して所与の要素qについて得られたパッチは、所定の基準である類似distに関連してRから抽出されたqの近傍要素のセットである。SASHを構築するための説明している実施の形態においては、データベース内の各ノードは、その階層レベルでラベル付けされている。また、各ノードについてのパッチは、所定の固定されたサイズとされ、かつ、同一のレベル以上のすべてのノードのセットに対して算出される。本発明においては、ノードごとのパッチを構築して格納するのに限定されるわけではなく、また別のノード・セットに関連する追加のパッチを構築して格納してゆくこともできる。

【００４３】

図３は、SASH構造の構成を本発明にしたがって生成されるパッチの構造と共に示した例示的な実施例である。図３に示されるように、パッチにより参照されるベクトル・ノードは、本質的には、基準のベクトル・ノードのレベル以上のSASH階層のいずれかに帰属される。加えて、これらの階層レベルにおけるノードの間から、パッチは、所定の“尺度距離”にしたがって、基準ノードに最も近い複数

のノードを含んで構成されている。この基準ノードは、クラスターの大域的な構成を与えるために階層構造のいかなる、またすべてのノードから選択することができ、また本発明の別の実施の形態では、基準ノードは、クエリーに関する基準ノード、すなわち検索されたドキュメントに特に関連したクラスター情報を与えるように、ユーザ入力クエリーを使用して決定することができる。基準ノードは、図３において、☆により示されており、パッチ内のノードは、図３のようにユーザ・クエリーに関連して整列されている。このパッチ構造はまた、詳細には後述するシステム内の適切なメモリ領域に格納されている。本発明においては、これらのパッチはさらに、後述するようにコンフィデンスにより関連づけられる。

【００４４】

＜コンフィデンスの計算＞

本発明の方法では、情報検索と“パッチ・モデル”として本明細書おいて参照する随伴規則の発見との双方を利用したクラスター化の新たなモデルを使用する。パッチ・モデルは、ドメインへの（非）類似近似のある種の尺度にしたがって、データのクラスターがデータ・セットからの要素に基づく近傍クエリーの結果として表現することができるということを仮定する。より、定式化すれば、Sをあるドメインから抽出された要素のデータベースであるものとし、ここで、”dist”を、D上に定義される対の間の上述した定義により与えられる尺度を満足する距離関数であるものとする。ここでさらに、RをSのサブセットとする。所与のいかなるq∈Dのクエリー・パターンに対しても、distによるRから抽出されたqに対するk－最近傍点のセットをNN(R，q，k)とし、下記の条件で選択されたものとする：

もしも、q∈Rの場合には、NN(R，q，1) = {q}とする。すなわち、もしqがデータ・セットのメンバーであれば、qをその最近傍点として考慮するものとする。

【００４５】

NN(R，q，k-1) ⊂ NN(R，q，k)（すべての1＜k≦|R|に対して）とする。すなわち、qのより狭い近傍は、より広い近傍に厳密に包含されるものとする。

【００４６】

これらの条件は、qがRにおける１つ以上のk－最近傍点セットを有する可能性

を考慮するためである。固有的に決定されるセットNN(R，q，k)は、qのk番目のパッチとして参照される（Rについて）か、またはqのパッチのうちの１つとして参照される。

【００４７】

図４は、データベースのパッチ（７－パッチ、１２－パッチ、および１８－パッチ）の構成を示した図である。破線で示した円は、ドキュメント空間の全体を表す。

【００４８】

ここで、R内において可能性のある２つのクラスターが、２つのパッチC_i = NN(R，q_i，k_i) and C_j = NN(R，q_j，k_j)として表されるものとする。C_jとC_iとの間の関連性は、AgrawalおよびSrikantら（前掲）により提案された随伴ルール発見に類似するナチュラル・コンフィデンス尺度にしたがって評価することができる。

【００４９】

$CONF(C_i,C_j) = |C_i \cap C_j|/|C_i| = |NN(R, q_i, k_i) \cap NN(R, q_j, k_j)|/k_i$.

すなわち、コンフィデンスは、C_iを構成する要素であって、またC_jを構成する要素に比例するものとして表現することができる。コンフィデンス値が小さい場合には、候補C_jは、C_iに対してわずかな影響しか与えないか、またはまったく影響を与えない。一方で、その割合が大きいと、C_jは、C_iに強く関連し、これを包含する可能性もある。

【００５０】

図５は、それぞれ８ベクトルおよび１０ベクトルを有するクラスターAとBとに対する関数CONFの本質を示した図である。２つのベクトルが、共にAとBとの交わりに存在し、したがって関数CONFが、A,Bの順でパッチに対して適用される場合、すなわちCONF(A,B)は、0.25、すなわち25%を与える。関数が(B,A)の順で適用される場合、すなわちCONF(B,A)の場合には、結果は0.2すなわち20%として与えられる。関数CONFは、データベースに共に含まれるサンプルから抽出される２つのパッチに対して適用することができる。

【００５１】

コンフィデンスの尺度はまた、共有近傍点の距離尺度の例としても捉えることができる。しかしながら、共有近傍点の情報への使用方法は、凝集クラスター化方法の使用方法とは本発明においてはきわめて異なるものである。凝集法は、上述した尺度を２つのパッチを統合するべきかどうかを判断するために使用するが、本発明において提案される方法は、尺度を２つの判断するべきパッチの間の随伴性のレベルの質を判断するために使用する。

【００５２】

＜クラスター内随伴性の計算＞

パッチ内における随伴性の自然な評価はまた、コンフィデンスの考え方で可能である。ここで、$C_q = NN(R, q, k)$を、パッチのクラスター候補とする。ここで、パッチC_qを構成するパッチは、$v \in C_q$のすべての要素について$C_v = NN(R, v, k)$の形のパッチのセットであるものと定義する。C_qが高い内部随伴性を有している場合には、C_qと、その構成パッチとの間に強い関連性を充分に予測することができる。他方で、内部随伴性が低い場合には、C_qとその構成パッチとの間の関連性は弱いことが期待される。したがって、セルフーコンフィデンスにおけるパッチのクラスター候補における内部随伴性が得られ、これを構成パッチに関連して候補パッチの平均コンフィデンスとして定義することができる。

【００５３】

$$SCONF(C_q) = (1 / |C_q|) * \Sigma_{v \in Cq,\ |Cv|=|Cq|}\ CONF(C_q, C_v)$$
$$= (1 / k^2) * \Sigma_{v \in Cq}\ |NN(R, q, k) \cap NN(R, v, k)|.$$

１のセルフーコンフィデンス値は、クラスターのすべての要素にわたり完全な随伴性があることを示すが、これが０に近づくにつれて内部随伴性は小さくなるかまたはまったく随伴性のないことを示す。

【００５４】

＜クラスター間のコンフィデンスを使用したクラスター境界の決定＞

ここで、対象としているノードqが、我々が評価を希望するR内のあるクラスターに随伴するものと仮定する。セルフーコンフィデンスの考え方を使用して、プロセスは、関心のある範囲$a \leqq k \leqq b$でこのクラスターを最も良く記述するqを元にしたkーパッチを決定する。理想的なパッチは、クラスター要素を主要に含むもの

であり、相対的に高いセルフーコンフィデンスを有するものであるが、パッチがより大きくなると、クラスターの外部の多くの要素を含み、比較的低いセルフーコンフィデンスを有することが予測される。２つのパッチに対する評価について焦点を当てる。サイズがkの内パッチ$C_{q,k}$ = NN(R，q，k)は、候補パッチのクラスターを示し、サイズがϕ(k)＞kの外パッチ$C_{q,\phi(k)}$ = NN(R，q，ϕ(k))は、内パッチの判定に対する適切さに対する局所的なバックグラウンドを与えるものとする。

【００５５】

所定の選択したkに対して、外パッチのそれぞれの要素の近傍点セットを評価する。近傍点の対(v,w)を考えるものとし、vが外パッチに属し、wが外パッチNN(R，q，ϕ(k))の要素であるものとする。vはまた内パッチに属し、wが内パッチNN(R，v，k)の要素であるものとすると、ここで、(v,w)は、内側近傍点の対である。

【００５６】

ここで、wが、外パッチの要素であれば、対(v,w)は、外パッチのセルフーコンフィデンスに寄与し、qを元にしたクラスターのデスクリプターとして内パッチの選択を妨げることになる。wがまた、内パッチの要素である場合には、(v,w)は内側対であり、したがってこの対は、内パッチのセルフーコンフィデンスに寄与することにより、vとqとの間の随伴性を強化することになる。

【００５７】

本質的には、qを含むクラスターをもっとも良く記述するkーパッチは、下記のようにして得られる。

ｉ）内パッチのセルフーコンフィデンスに寄与する内側ペアを高い割合で含むこと、および

ｉｉ）外パッチのセルフーコンフィデンスに寄与しない近傍点の対（内側対である必要はない）を高い割合で含むこと、である。

【００５８】

ｉ）の種類の高い割合は、kーパッチ内での随伴性の高いレベルを示すが、ｉｉ）の種類の高い割合は、局所的なバックグラウンドとの間で高い差別性を有し

ていることを示すものである。両方の考察は、等しく重要なので、これらの割合は、別々に考慮されるべきである。上述した考察は、a≦k≦bの範囲の選択されるすべてのｋにわたり、２つの割合SCONF($C_{q,k}$)および1 - SCONF($C_{q,\phi(k)}$)の和が最小となるようにして考慮される。

【００５９】

相対セルフーコンフィデンス(RSCM)の問題は、後述するようにして定式化することができる。

【００６０】

$\max_{a \leqq k \leqq b}$ RSCONF($C_{q,k}$, ϕ),

上記式中、

【００６１】

RSCONF($C_{q,k}$, ϕ) = SCONF($C_{q,k}$) - SCONF($C_{q,\phi(k)}$)

= SCONF(NN(R, q, k)) - SCONF(NN(R, q, ϕ(k)))

であり、RSCONFは、ｋーパッチ$C_{q,k}$のRおよびϕに関する相対セルフーコンフィデンスとして参照される。最大を与えるｋーパッチは、この領域にわたるqのクエリー・クラスターとして参照されるものとなる。RSCMは、最尤見積もり(maximum likelihood estimation)の形式として考えることができ、これは、近傍点の対がクエリー・クラスターとして内パッチを選択することを指示するか、または指示しないかを分類するものである。

【００６２】

図６は、クエリー要素のパッチ・プロファイルの一部として本発明の方法に含まれるSCONFの計算を実行するための擬似コードのサンプルを、近傍点リスト、NN(R, q, ϕ(b)) and NN(R, v, ϕ(b))がすでに、すべてのvについて利用可能であるものと仮定して示した図である。SCONF(NN(R, q, k))を直接的に算出するのではなく、SCONF(NN(R, q, k))は、１つのアイテムによりパッチを展開し、差分的に得るべく計算を行うことによって、SCONF(NN(R, q, k-1))から得られる。

【００６３】

本発明においては、説明するRSCM方法は、外パッチのサイズがk（kは整数である。）の値に依存する仕方で多くの変形例を可能とする。ϕ(k)=2kとする簡単な

選択が、内パッチに関連して外パッチの要素であるか、要素でないかということについて最良のバランスを与えるものの、φ(k)の選択にあたっては他の考察も影響を与える。例えば、境界を明確にするための値を算出するためのコストからは、最大パッチ・サイズとしてm＜2bを使用することが好ましい。この場合、外パッチ・サイズは、φ(k)＝min{2k,m}とし、外パッチ・サイズと内パッチ・サイズとの間のm/bの最小値が、依然として１よりも実質的に大きくなるように選択することができる。

【００６４】

本発明においては、RSCM法の設定は、内側のクラスターの随伴性が外側の差と同程度に重要であることを仮定する。しかしながら、本発明においては異なる重み付けを相対セルフーコンフィデンスの値への内側及び外側の寄与に対して与えることもできる。すなわち、下記式、

【００６５】

$RSCONF'(C_{q,k}, \phi) = w_1\ SCONF(C_{q,k}) - w_2\ SCONF(C_{q,\phi(k)})$,

の関数を最大化する代わりに、重み付け０＜w_1および０＜w_2の実数値を選択することもできる。

【００６６】

この段階で、格納されたそれぞれのパッチ$C_{q,m}$ = NN(R, q, m)は、1≦k≦mの範囲のkのすべての値に対して、$C_{q,m}$のそれぞれの副パッチ$C_{q,k}$=NN(R, q, k)それぞれに対する、セルフーコンフィデンス値のリストを伴っている。SCONFとして以下に参照するデータ構造を図７に示す。このデータ構造は、ハードディスクまたはメモリといった適切な記憶手段内に格納され、本発明のクラスター選択機能により参照される。

【００６７】

本発明のさらなる別の変形例は、計算のコストを節約するものである。RSCONF値を計算するコストは、外パッチのサイズが増大するに連れて二次的に増加する。このコストは、データ・セット全体への直接的なRSCM法の実際的な適用において見出されるクラスターのサイズを制限してしまうことになる。しかしながら、これらの制限は、ランダム・サンプリング技術を使用して回避することができる

。RSCM問題を解くための範囲をa≦k≦bに制限するように調整することで大きなクラスターに適応させる代わりに、サイズを変化させたデータ・サンプルを集めるのに対して、固定された範囲においてパッチ・サイズの検索することができる。

【００６８】

上述した変形例を理解するために、ある大きな値cについてR ⊆ Sの等しくランダムなサンプルと仮説的なクエリー・クラスターNN(S，q，c)との間の関係を考察する。NN(S，q，c)とRとの交わりは、パッチNN(R，q，k)を生成する。ここで、k = |NN(S，q，c) ∩ R|である。パッチNN(S，q，k)は、Rにおけるqのクエリー・クラスターとしてのサンプルR-NN(S，q，k)の選択値に関連して、NN(S，q，c)のプロキシを提供し、これは、全体のデータ・セットに関してqに対してのクエリー・クラスターとしてのNN(S，q，c)の適切さの指標とすることができる。

【００６９】

a≦k≦bの場合には、プロキシ・パッチをRSCM法により見積もることができる。それ以外には、kは、aとbとの間にはなく、パッチを見積もることができない。未知の“真の”クラスター・サイズcについては、評価されないプロキシ・パッチの可能性の限界は、MotwaniおよびRaghavan(R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press, New York, USA, 1995.)の標準的なChernoff境界技術を使用して得ることができる。

$E[k] = \mu = c\,|R| / |S|$

$Pr\,[k < a \mid c] \leqq e^{-\mu}\,[e\mu / (a-1)]^{a-1}$

$Pr\,[k > b \mid c] \leqq e^{-\mu}\,[e\mu / (b+1)]^{b+1}$.

これらの境界は、近似的なサイズの収集に加え、aおよびbの近似的な値を選択するためのガイドとして使用することができ、充分に大きなｃについての所望する確率について、少なくとも一つのサンプルを、少なくとも１つのプロキシ・パッチがaとbとの間のサイズとなるようにすることができる。

【００７０】

例示的な実施例としてh、均等なランダム・サンプルである$|R_i| = |S| / 2^i$ f

or i ≧ 0である{R_0, R_1, R_2, ...}について考察する。ここで、NN(R_i, q, k_i)を、NN(S, q, c)のプロキシ・パッチとする。ここで、cは、少なくとも25以上であることが保証された未知の数である。a=25,b=120の制限を選択する場合には、少なくとも一つのサンプルR_iに対して、i番目のプロキシ・パッチの所望するサイズ$\mu_i = E[k_i]$は、$44 \leqq \mu_i \leqq 88$の範囲内に存在する。上述した境界を適用すると、μ_iがこの範囲に制限される場合には、NN(R_i, q, k_i)がRSCM法により評価できない確率は、低いと見積もられる（0.004285より小さい）。

【００７１】

言い換えれば、この範囲とサンプルとを選択すると、まったく評価されないプロキシ・パッチの確率は、1/233以下とすることができる。このエラー境界は、まったく従来と同様のものであり、失敗する確率は、実際的にはきわめて低いものである。

【００７２】

RSCM法がプロキシ・パッチNN(R_i, q, k_i)を、クラスター評価として提供する場合であっても、Sにおけるクラスターに対応するサイズを推定するための正確な方法はない。しかしながら、後述する最大蓋然性評価の原理、$E[k] = k_i$でのc = $E[k]$ |S| / |R_i|の値は、真のクラスター・サイズの自然な見積もりを構成する。サンプルR_iについて見積もることができる最も小さなクラスター・サイズは、したがって、(a |S|) / |R_i|となる。

【００７３】

同一のクラスターがいくつかの異なるサンプルにわたって検出される場合には、真のクラスターのサイズの見積もりは、妥当でないことに留意されたい。にもかかわらず、実際には大きなRSCONF値は概ね、クラスターのサイズが正確に決定されない場合でさえも、クラスターの存在を示す信頼性のある指標となることが判明した。

【００７４】

＜要素の再分類＞

本発明においてはさらに、共通のクエリー要素へのメンバーの近接性の利点により、RSCM法で生成されたクラスターは、テキスト・マイニングにおいて望ましい

特徴である凝集クラスター化法により生成されるクラスターよりも、より凝集性を有する傾向にあることが見出された。現実的には、クエリー・クラスターは、ペア間の距離尺度よりは、球形の形状に向かって行くことになる。

【００７５】

RSCM問題のためのクラスター・パッチの解は、全体として同様のクエリー要素に基づく他の方法に比較して高いレベルの相互随伴性を示すものの、上述したクラスターの要素は、クエリー要素それ自体に伴なわれても、伴われなくとも良い。むしろ、クエリー要素は単に、相互に良好な随伴性を有する近傍データを見出すためのスタート点を与えるにすぎない。クエリー要素がその随伴クラスターに比較してアウトライアーであるか、または見出されたクラスターの実質的な部分がアウトライアーを構成するように思われるような別の状況の場合には、第２のクラスター化基準にしたがって外パッチの要素を再評価することが効果的である。このような再評価は、より球形である凝集クラスターの発見を可能とする。

【００７６】

多くの方法によりクエリー・クラスターの近くで要素を再分類することが可能となる。図８には、このような変形例を説明した疑似コードの記述を示す。図８に示したプロセスは以下のようにして実行される。

ｉ）元のクエリー・クラスターを決定した内側のk－パッチを与え、すべての対応する外側パッチの要素をクエリー要素において共有されるk－近傍点の数にしたがって再評価を行う。具体的には、すべての$v \in NN(R, q, \phi(k))$を、コンフィデンス値$CONF(C_q, C_v)$にしたがって、最高から最低へとランク付けを行うqからの距離に応じて連結は破られることになる）。ここで、$C_q = NN(R, q, k)$であり、$C_v = NN(R, v, k)$である。

ｉｉ）最も高い方からk個の要素を、新たな調整されたクラスターとして報告する。これとは別に、外パッチの要素の全体のランキングを報告しておき、ユーザが最終的なクラスターの要素の判断を残しておくこともできる。この方法においては、元の内パッチの外側の要素であって依然として外パッチの内側にある要素が新たなクラスターに含まれるものとして選択され、近傍点間の元々のパッチの要素を多く有することが示される。

【００７７】

＜クラスターの選択＞

提案される全体的なクラスター化の方法であるPatchCluster機能は、均等なランダム・サンプル{R_0, R_1, R_2, ...}の収集から得られるクエリー・クラスター相互関係(query cluster relationship:QCR)のグラフを構築する。ここで、すべてのj＜iおよび$|R_i| = ceil(|S|/2^i)$ for $0 \leqq i < \log_2 |S|$について、$R_i \subset R_j$である。このグラフは、コンフィデンスに類似するいくつかのパラメータに依存し、これを支持するための随伴規則生成のしきい値は、下記の通りである。

ｉ）（クラスター品質）クラスターの相対セルフーコンフィデンスについての最小しきい値α；

ｉｉ）（クラスター差）概ね同一のサイズの２つのクラスターの間のコンフィデンスの最大のしきい値β（共通のサンプルRiから得られる）；

ｉｉｉ）（随伴品質）随伴クラスターの間のコンフィデンスの最小のしきい値γ（共通のサンプルから得られるものである必要はない）；

ｉｖ）（随伴スケール）２つの随伴するクラスターの間のスケールの差の最大のしきい値δ（すなわち、差|i-j|であり、R_iとR_jとは、クラスターを導出するサンプルである。）

図９は、本発明に使用されるPatchCluster法を記述した擬似コードのサンプルである。基本的なQCRの構成手法は、下記のようにまとめることができる。

【００７８】

１．QCRノード・セット：

それぞれ$0 \leqq t < \log_2 |S|$について、サンプルR_tの要素から、R_tの異なるクエリー要素に基づくそれぞれ$C_i = NN(R_t, q_i, k_i)$であって、$a \leqq |C_i| \leqq b$のクラスターからなるクエリー・クラスター$QC_t = \{C_1, C_2, ..., C_{|Rt|}\}$を収集する。RSCONFにしたがって使用できるクエリー・クラスターの中から、可能な限りQC_tの要素を選択する。この場合、i＜jであれば、$RSCONF(C_i) \geqq RSCONF(C_j)$とし、このための条件として、下記の２つの条件を適用する。

ｉ．（クラスター差）すべての$1 \leqq i < j \leqq m_t$について、$\max \{CONF(C_i,C_j), CONF(C_j,C_i)\} < \beta$とし；

ｉｉ．（クラスター品質）すべての$1 \leqq i \leqq |R_t|$について、$RSCONF(C_i) \geqq \alpha$とする。

これらのクラスターは、レベルｔでのQCRグラフにおけるノードとなる。

【００７９】

２．QCRエッジ・セット：

QCiにおける$i \leqq j \leqq i+\delta$であるような異なるクエリー・クラスター$C_i = NN(R_i, q_i, k_i)$および$C_j = NN(R_i, q_i, k_i)$のそれぞれのペアについて、$C'_j = NN(R_i, q_j, 2^{j-i}k_j)$として、$\max\{CONF(C_i,C'_j), CONF(C'_j,C_i)\} \geqq \gamma$の場合に、指示されたエッジ$(C_i, C_j)$,および$(C_j,C_i)$を、QCRグラフに挿入する。$CONF(C_i,C'_j)$および$CONF(C'_j,C_i)$の値をエッジ$(Ci, C_j)$,および$(C_j,C_i)$の重み付けとして適用する。

【００８０】

グラフのそれぞれのレベルは、クラスターのセットのレベル、aおよびbとに依存する領域内にある見積もりサイズからなる、粗い断面として見ることができる。それぞれのスライスにおいて、候補は、それらのRSCONF値にしたがって余すことなく選択される。また、新たな候補は、それらが以前に許容された候補から充分に異なる場合にのみ許容される。

【００８１】

本発明においては、共通のレベルにおける重複したクラスターの生成は排除されるものの、重複したクラスターが、異なるレベルにおいて生成する場合には許容される。QCRグラフは、このためわずかな時間だけのみ所定のクラスターを含むことができる。いくつかの生成されたレベルにおける同一のクラスターの存在は、共通する概念を共有する２つのクエリー・クラスターが、重なり合っているものと判断されるので、実質的に構造の連結性を向上させることとなり、エッジにより連結されることになる。図９は、本発明において“PatchCluster法”として参照される、クラスターを排除するための擬似コードのサンプルを示した図である。

【００８２】

αの値を低下または増加させることにより、ユーザは、グラフに出現するクラ

スター・ノードの数を増加または減少させることができる。βの値を増加させることはまた、クラスターの数を増加させることになるが、これは、所定のサンプルから１つ以上のクラスターにより個別的な概念が共有されてしまうというリスクをもたらす。クラスター化の規則を誘導する目的のため、高い連結性のあるグラフとすることが好ましい。QCRグラフの２つの随伴クラスターの間のスケールの差に対する最大しきい値δは、後述する理由から小さな固定値とするべきである。

【００８３】

PatchCluster法の別の変形例は、クラスターの数の制御を含むものである。上述したように、生成されるクラスターの数は、報告されたクエリー・クラスターの相対セルフーコンフィデンスしきい値αを特定することによって制御される。その代わりに、ユーザは、それぞれのデータ・サンプルについて別々のクラスターの数を決定するオプションが与えられても良い。所定のレベルｔに対して、このことは下記のようにして実行される。

ｉ）レベルtから得られたクエリー・クラスターの相対セルフーコンフィデンスについて最小のしきい値α_tを特定する。

ｉｉ）レベルtから得られるクエリー・クラスターの絶対数に付いての最大のしきい値を特定する。

【００８４】

クラスターの数についてのしきい値が与えられると、クラスターの選択は、所望するクラスターの数が得られた場合や、すべての候補を考慮した場合にどれが最初に発生しても停止される。

【００８５】

PatchCluster法では、PatchCluster/RSCMのパラメータは、上述した方法すなわちアルゴリズムが実装されるシステムに依存して決定することができる。決定されるパラメータは、下記のものを挙げることができる。

【００８６】

＜内パッチ・サイズの範囲＞

内パッチのサイズの範囲[a,b]は、この方法によって任意に大きなクラスターで

も発見することができるように選択されるべきである。aとbとを、失敗の確率を分析してより精度良く選択することも可能であるが（上述したChernoff境界を使用する）、下記の汎用的な原理を適用することができる。パラメータは、サンプルのサイズが小さいことによる変動を克服するに充分なだけ大きくする必要がある。変数aは、20以上であることが要求される。パラメータbは、得られるレベルについてターゲットとするクラスター・サイズの範囲が実質的なオーバーラップを有しているように選択される必要がある。このことは、bをaの約3倍以上とすることにより達成されることが見出された。

【００８７】

＜最大パッチ・サイズ＞

また、最大パッチ・サイズは、効率の理由からできるだけ小さく選択することが必要である。しかしながら、最大パッチ・サイズは、実質的にbよりも大きくなるように選択される必要があり、$\phi(b)=2b$として選択することが理想的である。しかしながら、$\phi(b)=1.25b$を選択しても、また良好な結果を与えることができることが見出された。本発明の最良の実施の形態においては、a=25、b=120、$\phi(k)=\min\{2k,150\}$とすることで、多くのデータ・セットについて満足する結果が得られるので、好ましいことが見出された。

【００８８】

共通のサンプルからのいかなる２つのクラスターの間のコンフィデンスについての最大のしきい値βは、データ・セットにかかわらず概ね0.4に設定される必要がある。実験によれば、共通のサンプルから重なり合うクエリー・クラスターは、近似的にまたはわずかに重なり合ういずれかの傾向にあることが示された。PatchCluster法により与えられるクラスター化は、bの正確な選択には比較的感受性は高いものではない。

【００８９】

＜しきい値δ＞

QCRグラフの２つの随伴するクラスターのスケールにおける差の最大のしきい値δは、いくつかの理由から常に小さな固定値とされる必要がある。大きな値は、最大のクラスターが圧倒的な数のきわめて小さなクラスターへと連結されるグラ

フを与える。この結果、QCRグラフは、ユーザを誘導するのについて、きわめて困難性をもたらす。レベル0からのすべてのクエリー・クラスターについては、式$NN(R_i, q_j, 2^{\delta} k_j)$の近隣を算出する必要がある。スケーラビリティを保証するために、δは、小さな定数として選択されなければならない。この値を、実験においてはδ=4とすることで、概ね最大で2^4から2^5の程度の差のサイズのクラスター間で随伴エッジを生成することができた。δについて上述のように選択することがきわめて好ましい。

【００９０】

後述するパラメータは、ユーザの特定の要求にしたがって設定することができる。

（ａ）クラスターの相対セルフーコンフィデンスに対する最小のしきい値α（または、これとは別に、それぞれのサンプル・レベルに対して最小のクラスターの相対セルフーコンフィデンスおよび／または所望するクエリー・クラスターの最大数）。0.1≦α≦0.2の範囲の値が好ましい。値をより小さくすると、クラスターの数が大きくなる。

（ｂ）QCRグラフにおける随伴クラスターの間のコンフィデンスについての最小のしきい値γ（共通のサンプル・レベルから得られる必要はない）。0.15≦γ≦0.2の範囲の値が好ましい。小さな値は、グラフのエッジの数をより多くする。

（ｃ）それぞれのクエリー・クラスターに適用されるキーワード・ラベルの数。

【００９１】

PatchCluster法のさらなる変形例は、正確な近傍パッチを算出するのではなく、近似的な近傍点パッチを計算するものである。PatchCluster法により実行される近傍点計算は、データ要素の数が多く、正確な近傍の情報を探索する場合には、高コストなものとなる。この方法の効率を改善するためには、近似的な近隣情報で置き換えることができる。SASHといった類似検索構造は、高い精度でシーケンシャル検索よりも高速にこの情報を生成するために使用される。

【００９２】

さらなるPatchCluster法の変形例は、ドキュメントーキーワード・ベクトルおよびキーワード・ベクトルの次元削減を実行するものである。

【００９３】

基本的なPatchCluster法は、図９に示すようにテキスト・データに適用する場合、ドキュメントが適切な重み付けを使用してベクトルとしてモデル化されていることを仮定する。キーワード空間が大きくともドキュメントあたりの平均の数が小さい場合には、ベクトル間の距離計算は、ベクトルが本質的、すなわち、非ゼロのエントリだけで、しかもその位置が格納されているのであれば効率的に実行される。しかしながら、ドキュメントあたりのキーワードの平均の数が大きな場合は、距離比較のコストを制限するためにしばしば次元削減が実行される。ドキュメントあたりの元の平均のキーワード数にかかわらず、LSIまたはCOVといった次元削減技術で、所望される場合にはクラスター化の前に適用することができる。発明の実施の形態の欄において与えられる実験結果は、次元削減の使用および非使用について、それぞれの効果を示すものである。

【００９４】

PatchCluster法のさらに他の変形例は、QCRグラフの簡単化に含ませることができる。PatchCluster法により与えられるQCRグラフは、多数のクラスターの対についての随伴情報を含む。しかしながらこの情報は、時として簡略化することなしにユーザを容易に誘導するためには高い密度を与えることとなる。グラフを適切に簡略化することを可能にするいくつかの方法は、下記の通りである。

【００９５】

ｉ）（レベル間の過渡的なエッジを排除する）例えば、グラフがu＜v＜wであるクラスター・ノード$C_1 = NN(R_u, q_1, k_1)$、$C_2 = NN(R_v, q_2, k_2)$、$C_3 = NN(R_w, q_3, k_3)$および随伴エッジ(C_1,C_2)、(C_2,C_3)、(C_1,C_3)を含むものと仮定する。エッジ(C_1,C_3)は、ユーザが(C_1,C_2)および(C_2,C_3)を介してC_1からC_2へと依然として誘導されるのでユーザから隠すことができる。

ｉｉ）（類似クラスターの省略）２つのクラスター$C_1 = NN(R_u, q_1, k_1)$および$C_2 = NN(R_v, q_2, k_2)$が、$CONF(C_1,C_2)$および$CONF(C_2,C_1)$の両方が充分に高い値を有するためにきわめて類似するものと考えられるときには、それらのうちそれぞれのノードを省略することができる。２つのノードのうち、一方のノードを保持し、他のノードを削除する（保持されるクラスター・ノードは、より高いRSCO

NF値を有すること、またはサイズが大きいことといった種々の仕方で選択することができる。）。削除されたノードを含むいかなるエッジであってもその後、保持されたノードに帰属され、例えばC_1が保存され、C_2が削除された場合には、エッジ(C_2, C_3)が、(C_1,C_3)に変換される。その結果いかなる重複したエッジでも削除されることになる。当然のことながら、他の簡略化方法も本発明においては採用することができる。

【００９６】

＜グラフィカル・ユーザ・インタフェイス：クラスターのラベリング＞

検索されたクラスターを表示させるための有用なグラフィカル・ユーザ・インタフェイスを提供するために、クエリー・クラスター・ラベリングおよび同定の問題を、テキスト・データおよびベクトル空間モデルの文脈においてここでは検討する。クエリー・クラスターは、単一のクエリー要素の限定された近傍内に存在するので、表示ベースのクラスター化のために、クラスターのデスクリプターとしてクエリーを使用することを試みる。しかしながら、クエリー要素は、クラスターの最良の代表となる必然性はなく、実際には、全体を記述する適切なクラスターの要素がない可能性もある。

【００９７】

クラスターに対してラベルを帰属させる１つの通常の方法は、ドキュメント・ベクトル・モデルにおいて使用されるような用語重み付け法にしたがい、クラスターのドキュメント内においてもっとも頻出する用語のランク付けされたリストを使用することである。用語にはそれぞれクラスターのすべてのドキュメント・ベクトルの全部にわたる用語重み付けに対応する合計（または平均に等価）に等しくなるようなスコアが与えられる。もっとも高いスコアを与える用語の所定数をランク付けして、ラベルとしてユーザに提供する。

【００９８】

COVやLSIといった次元削減技術を使用する場合には、元の次元削減されないドキュメント・ベクトルは、もはや利用可能ではないか、または格納および検索するのにはコスト的に高くつく。それにもかかわらず、元のベクトルなしに意味のある用語のリストは依然として抽出することができる。まず、i番目の用語は、z

$z_{i,j} = 1$, if $i = j$, and $z_{i,j} = 0$などのように、元のドキュメント空間の単位ベクトルzi= (zi,1, zi,2, ..., zi,d)とすることができる。ここで、クエリー・クラスターNN(R, q, k)に属するドキュメント・ベクトルの平均をμとする。この記述を使用して、ｉ番目の用語についてのスコアは、簡単に$z_i \cdot \mu$として記述することができる。しかしながら、$\| z_i \| = 1$であり、μは定数なので、これらのスコアにしたがう用語のランク付けは、下記の尺度にしたがうランク付けとなる。

【００９９】

$z_i \cdot \mu / \| \mu \| = \text{cosangle}(z_i, \mu) = \cos \theta_i$,

上記式中、θ_iは、ベクトルz_iとμとの間の角度を表す。

【０１００】

次元削減と共に、元の空間のベクトルvとベクトルwとの間の対の距離cosangle(v, w)は、次元削減された空間のvおよびwにそれぞれ等価な、v’およびw’を使用してcosangle(v’, w’)として近似することができる。したがって、ベクトルz_iおよびμの次元削減された相手方をそれぞれz'_iおよびμ'として、cosangle(z_i, μ)は、cosangle(z'_i, μ')で近似することができる。また、cosangle(z'_i, μ')は、クエリー・クラスターの次元削減されたベクトルの平均であるμ''を使用して、cosangle(z'_i, μ'')で近似することができる。ベクトルz'_iは、すべての$1 \leqq i \leqq d$に対してあらかじめ計算することで、用語のランク付けされたセットが効果的に次元削減されたアトリビュート・ベクトルを集めることでμ''に基づいて最近傍検索の手段により、生成することができる。dは、典型的には相当に小さいので、このような検索のコストは、クラスターの生成のコストに比較にして無視できるものとなる。

【０１０１】

次元削減したクラスターのラベリング法は、下記の通りである。

ｉ）i番目のアトリビュートについて、すべての$1 \leqq i \leqq N$に対して、次元削減アトリビュート・ベクトル$z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,d})$をあらかじめ計算する。ここで、Wを次元削減アトリビュート・ベクトルとする。

ｉｉ）$\mu'' = S_{v \in NN(R, q, k)}$ vを計算する。ここで、vおよびqは、次元削減

されたデータのベクトルである。

ｉｉｉ）λをクラスターの所望するラベルの数として、Wにおけるμ”のλ－最近点を、cosangle尺度の値を減少させながら計算する。

ｉｖ）クラスターのラベルとしてλ個の近傍のランク付けされたリストに対応するアトリビュートを取得する。これとは別に、cosangleそれ自体の値をユーザに対して表示させることもできる。また、さらに別に近似的な最近傍を、SASH、または別の類似検索方法を使用して生成されたように使用することもできる。

【０１０２】

パートＩＩ．情報検索のためのシステム

図１０は、本発明のアルゴリズムが実装されるシステムを示した図である。図１０に示されるように、システムは概ねコンピュータ１０と、ディスプレイ装置１２と、キーボード１４といった入力デバイスと、マウス１６といったポインタ・デバイスとを含んで構成されており、本発明にしたがい、ユーザが情報検索のためのクエリーを入力することができる構成とされている。コンピュータ１０は、また、検索されるドキュメントを格納するデータベース１８を管理していると共に、格納されたドキュメントをデータベースから検索する。コンピュータ１０は、LAN、またはWANまたはインターネットといった通信ライン２０へと、Ethernet（登録商標）、光通信、またはADSLにより、ハブまたはルータ２２を介して接続されている。

【０１０３】

通信ライン２０が、企業の複数のサイトを相互接続するLAN/WANおよび／またはインターネットであるものとすれば、コンピュータ１０は、クライアントおよび／またはユーザからの入力クエリーが伝送され、情報検索を実行するサーバとされても良い。サーバ・コンピュータ１０は、本発明のアルゴリズムによりドキュメントを受信したクエリーに関連して検索し、検索された結果を、クエリーを発行したクライアントへと返信する。当然ながら本発明は、上述した情報検索をインターネットを介した有料の情報サービスとして登録クライアントへと提供することもできる。これとは別に、コンピュータ１０は、特定の用途に対して好適となるように調整されたスタンドアローンのシステムとすることもできる。

【０１０４】

図１１は、コンピュータ１０内に実装される機能ブロックの詳細図である。コンピュータ１０は、概ねベクトル生成部２４と、SASH生成部３６と、SCONFリストを生成するためのコンフィデンス決定部３８と、パッチ規定部２６とを含んで構成されている。ベクトル生成部２４は、キーワード・リストまたは所定の規則を使用して、データベース１８に格納されたドキュメントから、ベクトル生成を実行し、生成されたドキュメントーキーワード・ベクトルを、対応するドキュメントへの適切なリンクまたは参照を可能とするように、メモリ、またはデータベースといった適切な記憶領域へと格納する。SASH生成部３６およびパッチ規定部２６は、本発明における近傍パッチ生成部３４を構成する。

【０１０５】

SASH生成部３６は、図２において示したアルゴリズムを使用してSASH構造を構築し、生成したSASH構造をメモリ領域３０へと格納させて、詳細には後述する処理を可能としている。SASH生成部３６は、コンフィデンス決定部３８が、CONF、SCONF、RSCONFといったコンフィデンス値を算出することができるように利用できる構成とされており、上述したアルゴリズムにしたがってSCONFリストを生成している。生成されたパッチ・データおよびコンフィデンス値は、図１０に示されるように、ハードディスク３２へと格納される。

【０１０６】

クエリー・ベクトル生成部４６は、検索条件およびクエリー・キーワードを受け取り、対応するクエリー・ベクトルを生成し、生成されたクエリー・ベクトルを適切なメモリ領域へと格納させる。上述したクエリーとしては、２つのタイプのものがある。１つは、すでに計算されてデータベース３２に格納されたクラスター構造を抽出するためのものであり、他のものは、未だ計算されておらず、格納されていないクラスター構造を検索するためのものである。ユーザ入力クエリー・ベクトルは、まず、検索部４０へと伝送される。説明している実施の形態では、検索部４０は、クエリーのタイプを解析する。クエリーが検索部４０に対してすでに計算され、格納されたクラスター構造を検索するように指示する場合には、クエリーは、メモリ領域３０に格納されたSASH構造に対してクエリーを適用

し、クエリーを受け取ったパッチ生成部４４がクラスター見積もり部２８へと検索されたデータを伝送する。クラスター見積もり部２８は、検索されたデータを受け取るとパッチ・データおよびそれに伴うSCONFリストを、ハードディスク３２から呼出し、クラスター内コンフィデンスSCONFおよびRSCONFと、クラスター間のコンフィデンスCONFとをそれぞれ使用してクラスターの見積もりを実行する。クエリーされたパッチ生成部４４において生成されたノードは、任意に選択されたノードであるか、またはユーザ入力クエリーにより検索されたノードである。

【０１０７】

得られたクラスター・データは、GUIデータ生成部４２へと伝送され、ディスプレイ部分（図示せず）のディスプレイ・スクリーン上にクラスター・グラフの構造をグラフィカルに表示させる。クラスター・グラフ構造の多くの表示の実施の形態が、本発明において可能である。代表的な１つの実施の形態では、クラスターを、クラスターの含む特徴的なキーワード（もっとも大きな数値）を水平方向に配置し、同時にクラスター・サイズの見積もりと共に垂直方向に配列するものである。このような表示がディスプレイ上に与えられる場合には、GUIデータ生成部４２は、パッチ格納部３２からのクラスター・データをソーティングし、ソートされたデータを適切なディスプレイ・バッファ（図示せず）といったメモリ領域またはコンピュータ１０の別の領域に格納する。

【０１０８】

本発明の特定の実施の形態では、検索部４０がクエリーがすでに検索されておらず、また格納もされていないクラスターの検索を指令していると判断した場合には、検索部４０は、SASHデータ３０を呼出し、ドキュメントーキーワード・ベクトルとクエリー・ベクトルとの間の類似を算出し、SASHの適切なノード・ベクトルを検索する。検索されたデータ・ベクトルは、その後、それら自信がSASHデータ３０内のクエリーのために使用されて、元のクエリーにより検索されるすべてのベクトルに対して類似ノード・ベクトルのリストを与える。類似ノード・ベクトルの類似のそれぞれのリストは、パッチ規定部２６へと送られ、その後コンフィデンス決定部３８へと送られて、パッチが生成される。これらのパッチは、

その後パッチ格納部３２へと追加される。検索されたパッチは、その後クラスター・見積もり部２８へとそれらの対応するSCONFリストと共に伝送されて、元のクエリーにおいて検索されたノードからなるクラスターが見積もられ、算出されたクラスター・データがGUIデータ生成部４２へと送られてクエリーの結果のグラフィカルな表示が行われる。

【０１０９】

GUIデータ生成部４２は、ソートされたクラスター・データをコンピュータ１０に直接接続されたディスプレイ装置（図示せず）へと送り、ディスプレイ・スクリーン上に検索されたクラスター・データの表示を行う。これとは別に、システムがブラウザ・ソフトウエアを使用してインターネットを介して検索された結果を提供する場合には、GUIデータ生成部４２は、クラスターの相関に関連するグラフィカル・データをブラウザ・ソフトウエアに適切な形式、例えばHTMLフォーマットで生成する。

【０１１０】

パートＩＩＩ．発明を実施するための実際のシナリオ

＜シナリオA－データベース内のノードの全体的なクラスター化＞

図１２は、データベース内に格納されたノードの全体的なクラスター化を実行する場合のシナリオのフローチャートを示した図である。シナリオＡのアルゴリズムは、まずドキュメントとキーワード・データとをステップＳ４０で読み込み、ステップＳ４２へと進んでドキュメントーキーワード・ベクトルとキーワード・リストとを生成する。アルゴリズムは、ステップＳ４４で上述したLSI法またはCOV法を使用して次元削減を実行する。その後シナリオAのプロセスは、ステップＳ４６で図２に示したプロセスにしたがって次元削減されたドキュメントーキーワード・ベクトルのSASHを構築する。図１２において示したアルゴリズムをステップ的に実行させるにしたがって生成されるデータ構造を、図１３に示す。

【０１１１】

一度SASH構造が構築されると、類似クエリーがその要素それぞれに対して実行されて、図１４（ａ）に示すように各ドキュメントの１つのパッチを生成する。シナリオＡのアルゴリズムは、その後ステップＳ４８で最適なパッチ・サイズお

よびRSCONF値を計算して図１４（ｂ）を与え、その後パッチをそれらのRSCONF値と共に図１４（ｃ）に示す構造として格納する。

【０１１２】

再度図１２を参照して、シナリオAのアルゴリズムは、ステップＳ５０へと進み、各SASHレベルにおいてβ=0.4以下のレベルのコンフィデンスの内パッチ随伴性のすべてに対してパッチの集団を選択する。その後、RSCONF値がα=0.15以上のパッチをさらに選択して、ステップＳ５２でクラスターを決定する。ステップＳ４６～Ｓ５２に関連するデータ構造を、図１５に示す。

【０１１３】

次いで、シナリオAのアルゴリズムは、ステップＳ５４へと進み、所定のしきい値γ以上の随伴コンフィデンス値を有するクラスター間の連結を行う。このデータ構造を、図１６に示す。これらの連結の結果は、クラスター・ラベルおよび対応するキーワードと共にステップＳ５６において、図１７に示されるようなグラフ表示として、グラフ的に与えられる。図１７ではシナリオAにしたがって生成されたクラスターのグラフの部分（γ=0.2）が示されており、その際には、COV次元削減法を使用した。図においてクラスター・ノード（長円で示した。）は、数x/yの対として示されており、xは、クラスターの見積もりサイズであり、yは、それに伴われる要素のパッチ・サイズである。キーワード・ラベルは、各クラスターに対して示されており、ボックスは、同一のラベルのセットを共有するクラスターのサブセットを連結している（ラベルの順番には、わずかな違いが可能である）。図１７では、106/53で示されるノードに対応するクラスターが示されている。このクラスターは、２つの大きなクラスターのセットの交差のニュース記事を含んでいるので特に興味が持たれるものであり、谷とそれらの開発および保存の記事の他、ゴミ用ダンプ車および他の埋め立ての記事を含んでいる。

【０１１４】

図１２に示したプロセスに含まれる詳細な手順は、下記の通りである。

ｉ）M個のドキュメントを、例えば、バイナリモデルまたはTF-IDF重み付けを使用してベクトルへとモデル化する。これらのベクトルの次元はNであり、これがデータ・セットのアトリビュートの数である。

ｉｉ）さらなる実施例としては、ベクトルのセットをNよりもきわめて小さな数（典型的には200～300）のベクトル・セットとなるように、COVまたはLSI次元削減法を使用して次元削減を適用する。次元削減を採用する場合には、また次元削減したアトリビュート・ベクトルを生成する。

ｉｉｉ）ｋ－最近傍点クエリーを適用するSASHを構築する。Stを、$0 \leqq t \leqq h$のｔ番目のSASHレベルとし、ランダムなサンプル$R_t = S_t \cup S_{t+1} \cup \ldots \cup S_h$とする（ここで、$S_0$は、SASHレベルの最低部とする。）

ｉｖ）$0 \leqq t \leqq h$のすべてについて$v \in S_t$である各要素について、$m = \phi(b)$の要素について近似的なｍ－最近傍点リスト（ｍ－パッチ）$NN(R_t, v, m)$を算出し、格納する。

ｖ）図１６に概略的に示したクエリー・クラスターのセットとクラスター構造とのグラフを算出する。

ｖｉ）次元削減が行われた場合には、セットのそれぞれのクエリー・クラスターについてクラスターを構成する次元削減されたドキュメント・ベクトルからアトリビュートであるキーワードのセットを生成する。

ｖｉｉ）好適なユーザ・インタフェイスを使用してユーザがブラウジングすることができるクラスターの得られたセット、それらのサイズ、ラベルおよびクラスター構造のグラフを作成する。

【０１１５】

＜シナリオB－個別的なクラスター：クエリー検索＞

シナリオBでは、シナリオAと同一の処理を使用してSASHを生成する。この後の本質的なステップを図１８に示す。また、シナリオBの処理において生成されるデータ構造を、図１９および図２０に示す。図１８に示されるように、シナリオBのプロセスは、ステップＳ６０でSASH構造を生成し、ステップＳ６２へと進んでユーザ入力クエリーqと、共にターゲットとするクラスター・サイズkとを受け取り、これらを適切なメモリ空間へと格納する。その後SASH内におけるノードを、ステップＳ６４においてSASH構造を使用してクエリーに関連して検索する。ステップＳ６４においては、SASHは、それぞれのランダム・サンプルR_tに関連して$0 \leqq t \leqq h$のすべてに対して１つの近隣パッチをクエリー要素qに関連して生成する

。その後、プロセスはステップＳ６６へと進んで、RSCONFを計算し、すべてのランダム・サンプルに対してRSCM問題をユーザ入力クエリーについて解く。書くサンプルに対して、クラスターは、これにより生成される。シナリオBのプロセスは、その後これらのクラスターを代表するラベルとキーワードとを、ステップＳ６８において与える。ステップＳ６４～ステップＳ６８において得られたデータ構造が、図２０に示されている。

【０１１６】

シナリオBの詳細な処理を下記に示す。

２－ｉ）シナリオAの手順をｉ～ｉｉｉまで繰り返す。

２－ｉｉ）ユーザに対してクエリー要素qおよびターゲット・クラスターのサイズkを要求する（データ要素に対しては必要ではない）。

２－ｉｉｉ）$t_a = \max \{t \mid k / 2^t \geqq a\}$および$t_b = \min \{t \mid k / 2^t \leqq b\}$を計算する。すべての$t_b \leqq t \leqq t_a$に対して、$NN(R_t, q, m)$を計算する。ここで、$m = \phi(b)$である。$v \in NN(R_t, q, m)$のすべてに対して、$NN(R_t, v, m)$を計算する。

２－ｉｖ）すべての$t_b \leqq t \leqq t_a$に対して、R_tについてのqに対するRSCM問題の解である$k(q,t)$を見出す。

２－ｖ）すべての$t_b \leqq t \leqq t_a$に対して、クエリー・クラスター$NN(R_t, q, k(q,t))$を構成する次元削減されたドキュメント・ベクトルからアトリビュートであるキーワードのセットを生成する。この手順については図１４において説明した通りである。

２－ｖｉ）得られたクラスターのセット、それらのサイズ、それらの対応するｍ－パッチのSCONFの値、およびそれらのクラスターのラベルを、ユーザに対して提供する。

【０１１７】

【実施例】

本発明を検討するために、本発明の方法を上述した２つのシナリオとして実装した。両方のシナリオについて、TREC-9テキスト検索実験の一部として利用可能なL.A.Timesニュース・ドキュメントのデータベースに適用することにより検討

した。データベースは、M=127,742ドキュメントからなり、これらからN=6590キーワード（アトリビュート）をアトリビュート・セットとして抽出した。有効性および汎用性を検討するために、データベースに対して次元削減を行わない手法と、次元削減(COV)を行ない手法の２つの手法を適用した。実装条件は、下記の通りであった。

（ａ）TD-IDF用語重み付けを6590個のアトリビュートに対して行った。

（ｂ）１つの実験セットにおいて、COV次元削減（6590から200へ）を行ない、他については次元削減を行わない。

（ｃ）ドキュメントの最近傍点の検索について、デフォルト・セッティングのSASH（親ノードのキャパシティp=4、子ノードのキャパシティc=16）を使用した。

（ｄ）アトリビュート・ベクトルの最近傍点の検索についても、デフォルト・セッティングのSASH（親ノードのキャパシティp=4、子ノードのキャパシティc=16）を使用した。

【０１１８】

それぞれのシナリオについて、パラメータφ、a、b、β、およびδを、次元削減に伴ういかなるパラメータ（削減次元であるdなど）と、または近似的な類似検索と同様にシステム管理者により設定できるものとした。

【０１１９】

実験条件は、以下の通りである。

（ａ）パッチ範囲のデリミター：a=25，b=120，$\phi(k)=\min\{2k, 150\}$

（ｂ）ドキュメントの最近傍点検索については、近似の精度に影響を与えることになるものの、時間スケーリングする因子として、

【０１２０】

$\mu' = 1.25$

$\mu = 1.25\ \phi(b)$

を用いた。すべての検索について、μ'個の近傍点を生成し、最も近い方からm個を使用した（μ'のより大きな値は、より高精度の結果を与えるものの、より検索時間を必要とする。）。

（ｃ）クラスターの相対セルフーコンフィデンスに対する最大しきい値を、$\alpha=0$

.15とした。

（ｄ）共通のサンプルからのいかなる２つのクラスターの間のコンフィデンスの最大のしきい値をβ=0.4とした。

（ｅ）QCRグラフにおける随伴クラスターの間のコンフィデンスの最小しきい値を、γ=0.15とした（共通のサンプル・レベルから得られる必要はない）。

（ｆ）QCRグラフの２つの随伴クラスターの間のスケールにおける差の最大のしきい値を、δ=4とした。

【０１２１】

計算アルゴリズムを、Java（登録商標）(JDK1.3)を使用して記述し、計算ハードウエアは、Windows（登録商標）2000を走らせた1GHzのプロセッサ速度および512Mbのメインメモリを搭載したIBM社製のIntelliStation（商標）E Proとした。

【０１２２】

２－１．実行時間および記憶コスト

RSCONF値は、一見すると計算するコストが高いように思われるが、注意深く実装することにより、コストを充分低くすることができる。これは、1～ϕ(b)の範囲のkについて、SCONF(NN(R，q，k))の値のプロファイルを効率的に計算することにより達成される。パッチ・プロファイルのプロットはまた、クエリー要素の近傍において随伴性の変動の効果的な表示を与える。

【０１２３】

後述する表には、シナリオAに伴う時間と、記憶空間のコストとをリストした。時間は、メインメモリに次元削減ドキュメント・ベクトルとアトリビュート・ベクトルとを読み込んだ時点から、クラスターの完全なセットおよびそれらのクラスター・グラフを算出するまでの計算に要した実時間の尺度である。クラスター化およびグラフ構築の時間的なコストは、すべての最近傍パッチがすでに計算されたものと仮定している。

【０１２４】

【表１】

Table 1

記憶のためのコスト（Mb）－次元削減の場合	
ドキュメントのSASH格納	30.1
キーワードのSASH格納	1.6
NNパッチの格納	161.6
次元削減ドキュメントの格納	204.4
次元削減キーワードの格納	5.3
格納全体	403

【０１２５】

【表２】

Table 2

時間的なコスト	次元削減なし	COV次元削減
ドキュメントのSASH構築時間（s）	460.7	898.8
キーワードのSASH構築時間（s）	--	26.6
全NN事前計算時間（s）	7,742.9	13,854.6
クラスター化とグラフ構築の時間（s）	126.2	81.8
全時間（s）	8,329.8	14,861.8
全時間（hr）	2.3	4.1

【０１２６】

２－２．近似的な最近傍点計算

後述する表は、ランダムに選択したSASHに対するサイズをm’として指定したクエリーについて、M個のドキュメントの完全なセットから近似的なm-最近傍点リストを見出すためのコストの平均を示す。比較のため、シーケンシャル検索を使用して正確なクエリーを実行させ、SASHの平均的な精度を算出した（取得されたリストにおける真の最近傍点の割合の尺度となる）。これらの値を使用して、シナリオBについての単一クエリー・クラスターの直接的な生成のコストを決定することができる。これら後者の見積もりにおいては、あらかじめ算出された最近傍点の情報がないドキュメントについてのSASHを使用するものと仮定した。

【０１２７】

【表３】

Table 3

SASH性能	次元削減なし	COV次元削減
平均SASHクエリー距離計算（s）	3,039.03	2,714.57
平均SASHクエリー時間（ms）	38.85	70.74
平均SASHクエリー精度（%）	62.93	94.27
正確なNNクエリー距離計算(s)	127,742	
正確なNNクエリー時間（ms）	1,139.19	2,732.5
単一クエリー・クラスターの距離計算（x 10^5）	4.59	4.07
単一クエリー・クラスター時間（s）	5.87	10.68

【０１２８】

２－３．クエリーによる全体のクラスター化

図２１には、COV次元削減を使用した例におけるパッチ・プロファイルの実施の形態を示す。このプロファイルは、シナリオAの方法により生成されたクラスターについてのものである。

【０１２９】

シナリオAにおいて次元削減の有無により生成されたクラスターの数を下記表に示す。

【０１３０】

【表４】

Table 4

見積りクラスター・サイズ（低－高）	次元削減なし	COV次元削減
6400 - 30720	1	1
3200 - 15360	1	2
1600 - 7680	8	8
800 - 3840	15	25
400 - 1920	32	50
200 - 960	70	84
100 - 480	206	135
50 - 240	405	216
25 - 120	760	356

【０１３１】

次元削減した実施例では、基本的な実施例に比較してより少数のマイナー・クラスターが見出されているが、より大きなクラスターも見出している。実験によれば、また次元削減を行う実施例では、より多くの相互連結を与えるクエリー・ク

ラスターのグラフを与え、キーワードの多様性をより解析することができた。

【０１３２】

本発明の方法は、コンピュータ実行可能なプログラムとして実装することができ、本発明のコンピュータ・プログラムは、C言語、C++言語、Java（登録商標）、またはいかなる他のオブジェクト指向言語においても記述することができる。本発明のプログラムは、コンピュータによりデータを書き込み読み込みが可能なフロッピー（登録商標）・ディスク、磁気テープ、ハードディスク、CD-ROM、DVD、光磁気ディスクなどに格納することができる。

【０１３３】

【発明の効果】

改善１：マイナー・クラスター

本発明のクラスター化方法は、通常のコンピュータにおいても良好な随伴性および良好に区別させつつ、データベース内の0.05%程度の小さなサイズのクラスターでも検出することを可能とする。この方法は、セット内におけるクラスターの数に関連して先験的な仮定を必要としない。また、この方法は、生成されるクラスターの局所的な影響のみしか考慮せずに済ませることを可能とする。重なり合うクラスターは、また許容される。これらの特徴は、従来の方法においてへ現実的でないものであるか、または不可能でさえあったマイナー・クラスターの発見を可能とする。

【０１３４】

改善２：クエリーに基づくクラスター化

本発明のクラスター化方法は、セットの完全なクラスター化の計算という過剰なコストなく、クエリーに近接する有意義な主要クラスターおよびマイナー・クラスターを効率的に生成することができる。本発明者が知る限りにおいて、大規模なテキスト・データに対して上述したことを可能とする方法は新規なものである。

【０１３５】

改善３：クラスター相関関係の自動的な決定

極めて僅かなクラスター化方法のみがクラスター間の重なり合いを可能とするに

すぎない。本発明の方法は、クラスターの重なりを使用して、クラスター間の対応を確立するので、ユーザを誘導する“クラスター・マップ”すなわち、関連した概念のグラフを生成することができる。関連性は、データのグループの間において階層の概念のようにではなく、むしろアトリビュート空間内での分類のようにして確立される。データ要素の重なり合いにしたがう組織化は、表現される概念をより柔軟なものとし、本発明の方法は、特に２つ以上の主要クラスターの交差部におけるマイナー・クラスターの発見を可能とする。

【０１３６】

改善４：クラスター品質の自動的な評価

RSCONF値およびパッチ・プロファイルは、クラスターを同定し、比較するばかりではなく、ユーザがクラスター内の随伴性のレベルおよび近接する要素との間の差異を評価するための手段となる。パッチの構造分布は、効果的により高次のテキスト・クラスターの可視化のための既存の空間表現方法を補うことができる。

【０１３７】

改善５：データ分布に対する知見の依存

ほとんどの分割に基づくアルゴリズムのようにではなく、本発明のクエリーに基づくクラスター化は、データの分布に対する知識や仮定を必要とせず、データが均一に分布しているかまたは分布に大きな変動があるかは問題にしない。これは、SASHが上述した特性を有しているように、最近傍リストの生成についても適用することができる。

【０１３８】

改善６：スケーラビリティ

SASH構造が近似的類似クエリーに使用される場合には、データ・セットSの全体のクラスター化のためにPatchCluster法に必要とされる漸近的な時間は、$O(|S| \log_2 |S| + c^2)$となる（Oは、定数である。）。ここで、cは、生成されるクラスターの数である（典型的には、|S|よりもかなり小さい）。先に出現するタームは、プロファイルを生成し、RSCONF値にしたがってクエリー・クラスターの候補をランク付けする際のコストである。重複したクラスターの排除およびグラフのエッジの生成は、合計で、$O(|S| + c \log_2 |S| + c^2)$の時間で実行することがで

きる。

【０１３９】

クエリー・クラスター構築のボトルネックは、最近傍点パッチの事前の計算にある。しかしながら、クラスター化は、近似的なクラスター境界および重なり合いを検出するためには完全に正確な最近傍点リストを必要とはしない。実際には近似的に正確な最近傍点リストを迅速に生成するためにSASHといった回避技術の一つを使用することで、さらにコスト効果が生じる。COV次元削減を使用したL.A.Timesニュース記事のデータ・セットについては、SASHは、シーケンシャル検索のほぼ95%の精度の精度で、粗々４０倍のスピードアップを達成する。パッチの事前計算のための漸近的な負荷は、$O(|S| \log_2 |S|)$で与えられるSASH処理の全コストにより支配される。

【０１４０】

これまで、本発明を図面に示した特定の実施の形態を使用して説明してきた。当然のことながら当業者によれば、多くの別の実施例、変更例、および／または付加例が開示された実施の形態に対して可能であることが理解され、したがって、本発明の真の範囲は、特許請求の範囲にしたがって決定されるものである。

【図面の簡単な説明】

【図１】　図１は、本発明のデータ構造を構築するためのフローチャートを示した図。

【図２】　SASH構造を構築するための処理の概略的なフローチャート。

【図３】　パッチ構造を含むSASH構造の概略図。

【図４】　本発明によるパッチ構造の例示的な概略図。

【図５】　コンフィデンス関数CONFの計算の代表例を示した図。

【図６】　SCONFの計算のための例示的な擬似コードを示した図。

【図７】　パッチおよびセルフ－コンフィデンスの格納構造を示した図。

【図８】　パッチ・プロファイルの更新を行うための例示的な擬似コードを示した図。

【図９】　PatchCluster法（パッチランク付けおよび選択を含む）の例示的な擬似コードを示した図。

【図１０】　本発明において典型的に使用されるコンピュータ・システムのブロック図。

【図１１】　本発明のコンピュータ・システムの機能ブロック図。

【図１２】　クラスターとそれらの相関関係のグラフを生成するための処理のフローチャート（シナリオA）。

【図１３】　図１２に示したシナリオAの処理に関連したデータ構造を図示した図。

【図１４】　図１２に示したシナリオAの処理に関連したデータ構造を図示した図。

【図１５】　図１２に示したシナリオAの処理に関連したデータ構造を図示した図。

【図１６】　クラスターの相関関係のグラフを示した図。

【図１７】　クラスターの相関関係の例示的なグラフ表示を示した図

【図１８】　クラスターとそれらの相関関係のグラフを生成するための処理のフローチャート（シナリオB）。

【図１９】　図１８に示したシナリオBの処理に関連したデータ構造を図示した図。

【図２０】　図１８に示したシナリオBの処理に関連したデータ構造を図示した図。

【図２１】　SCONF値と見積もられたクラスター・サイズとによりプロットされたプロファイルを示した図。

【符号の説明】

１０…コンピュータ

１２…ディスプレイ装置

１４…キーボード

１６…マウス

１８…データベース

２０…通信ライン

２２…ハブ／ルータ

２４…ドキュメント・ベクトル生成部

２６…パッチ規定部

２８…クラスター見積もり部

３０…メモリまたはデータベース（SASH記憶部）

３２…ハードディスク

３４…近傍パッチ生成部

３６…SASH生成部

３８…コンフィデンス決定部

４０…検索部

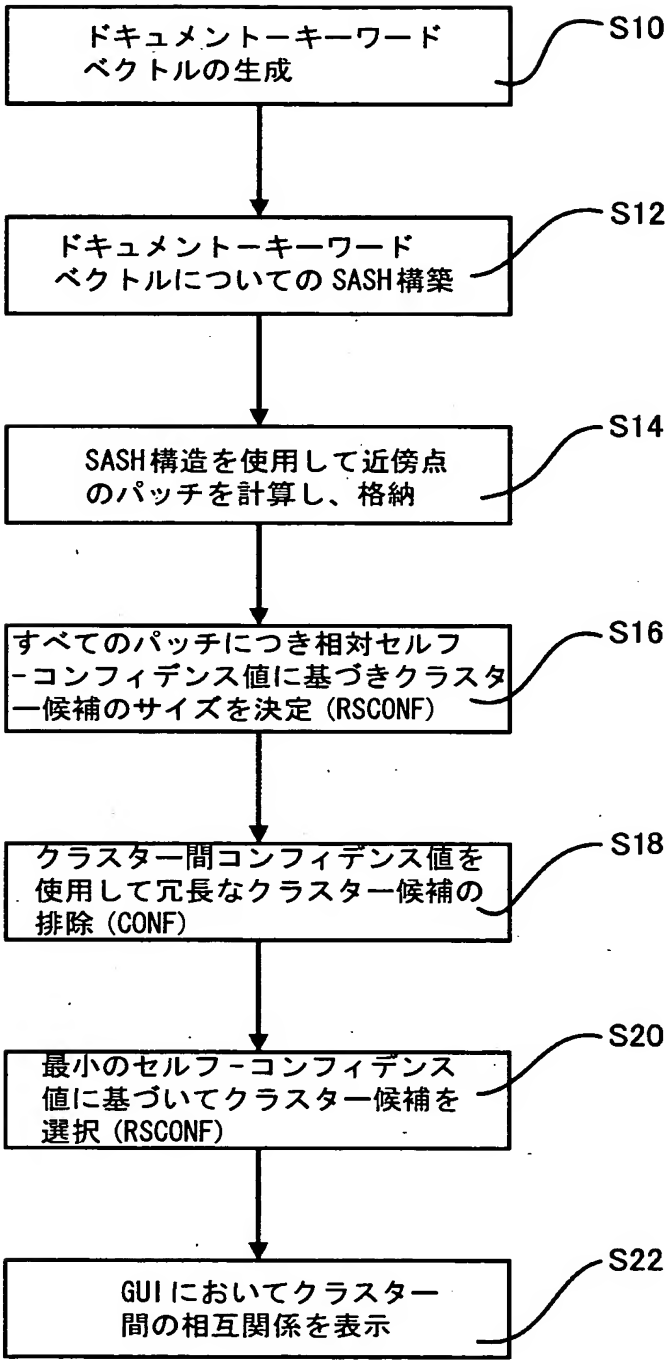
４２…GUIデータ生成部

４４…クエリー・パッチ生成部

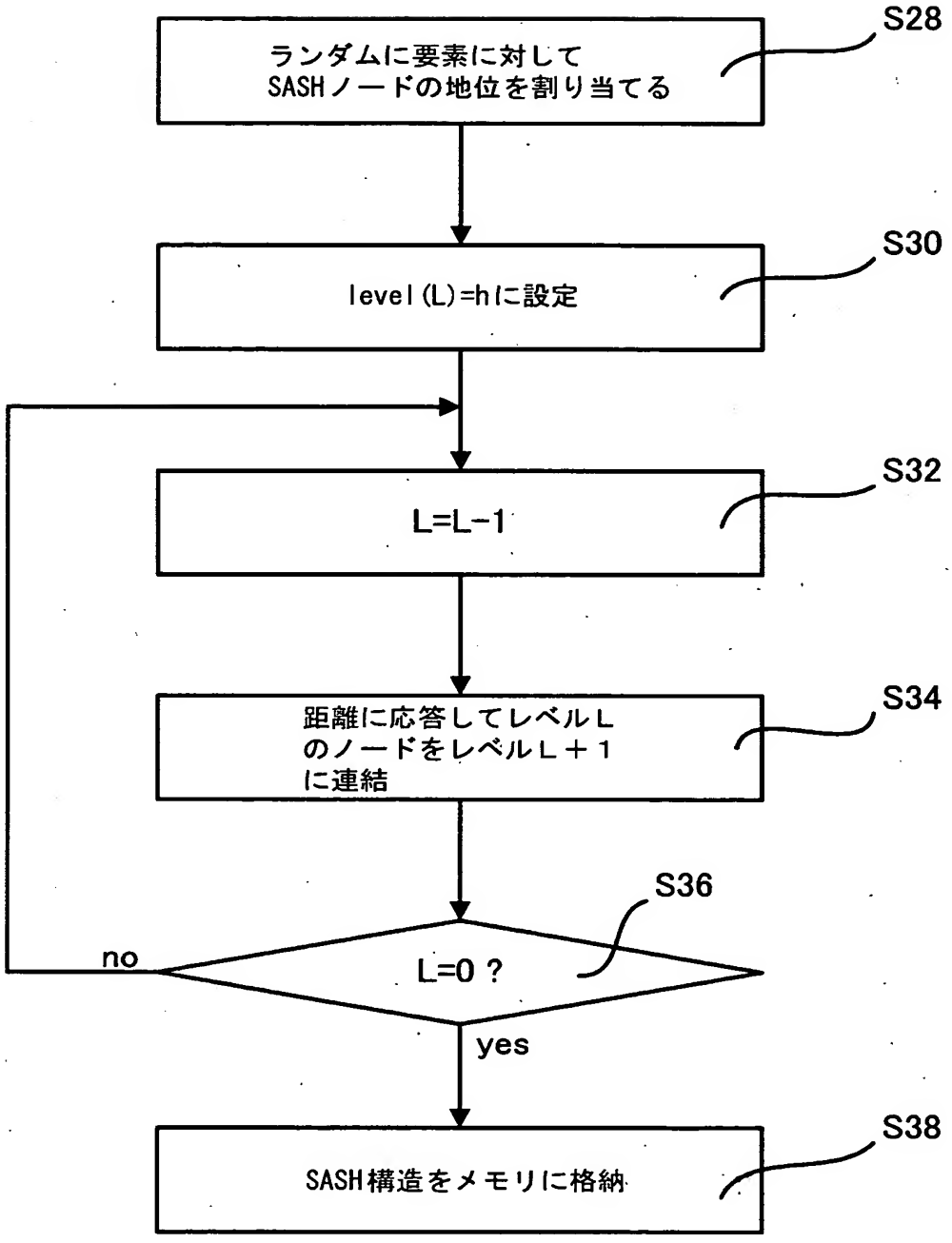
４６…クエリー・ベクトル生成部

【書類名】　図面

【図１】

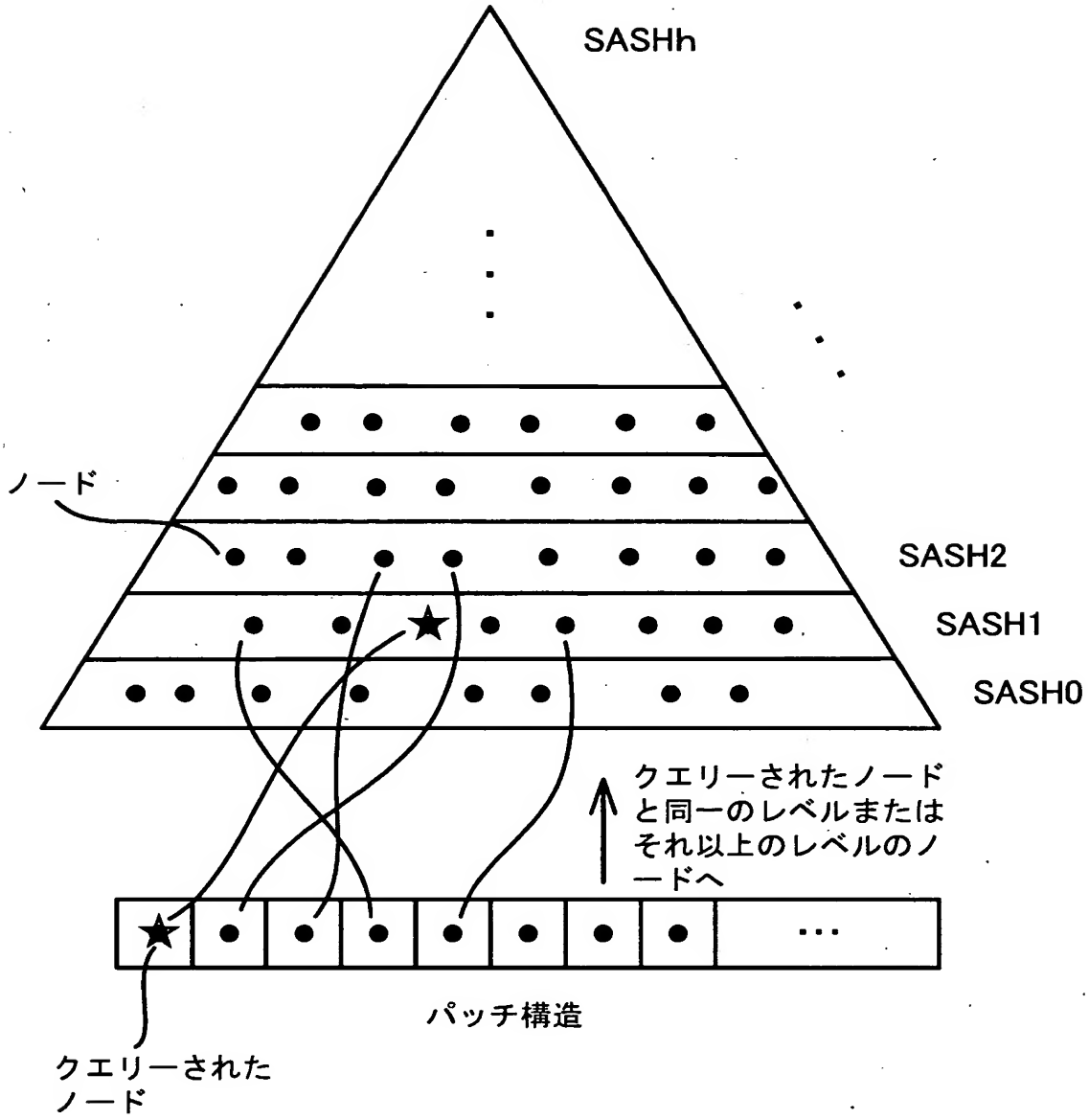

ドキュメントーキーワード
ベクトルの生成
S10
ドキュメントーキーワード
ベクトルについてのSASH構築
S12
SASH構造を使用して近傍点
のパッチを計算し、格納
S14
すべてのパッチにつき相対セルフ
-コンフィデンス値に基づきクラスタ
ー候補のサイズを決定（RSCONF）
S16
クラスター間コンフィデンス値を
使用して冗長なクラスター候補の
排除（CONF）
S18
最小のセルフ-コンフィデンス
値に基づいてクラスター候補を
選択（RSCONF）
S20
GUIにおいてクラスター
間の相互関係を表示
S22

【図２】

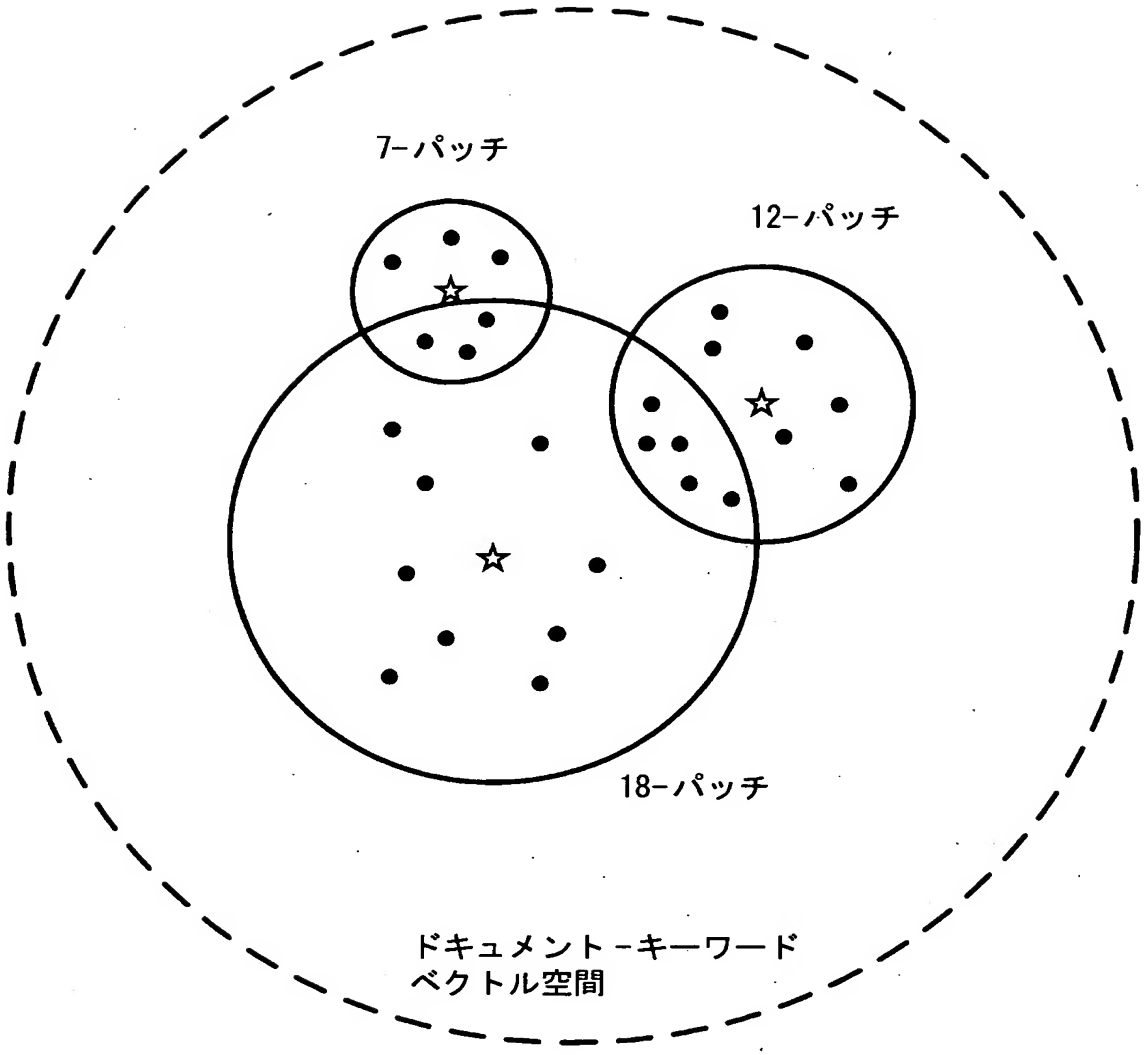


出証特２００３－３０３７０１４

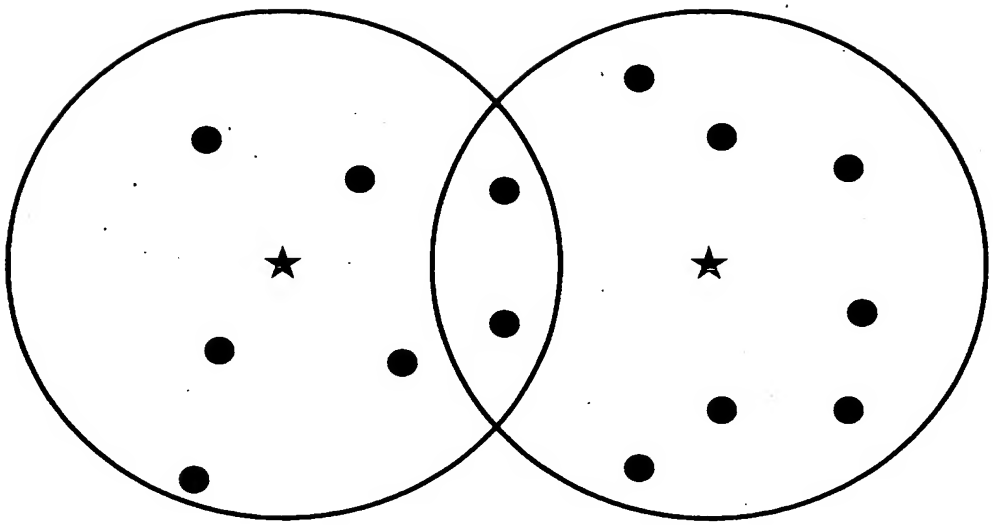
【図３】


SASHh
ノード
SASH2
SASH1
SASH0
クエリーされたノード
と同一のレベルまたは
それ以上のレベルのノ
ードへ
パッチ構造
クエリーされた
ノード

【図４】



【図５】



$CONF(C_i, C_j) = 2/8 = 25\%$

$CONF(C_j, C_i) = 2/10 = 20\%$

【図６】

```
Profile (query q; maximum patch size m): SCONF list SCONFL
    {Let QNL be the m-patch prec mputed for query q.}
    {Let NNL be a list of the m-patches precomputed for every element of QNL.}
    {Initially, w.count = ∅ is assumed for every element v in the data set.}
1.  score ← 0;
    {Initially, no query neighbors are in the current patch.}
    for i = 1 to m do
2.      QNL[i].count ← 0;
    end for
    for i = 1 to m do
        {Retrieve the number of times QNL[i] has been encountered as an external neighbor so far.}
3.      score ← score + QNL[i].count;
        {Indicate that henceforth QNL[i] is in the current i-patch.}
4.      QNL[i].count ← present;
        for j = 1 to i − 1 do
5.          w ← NNL[j, i];
            if w.count = present then
6.              score ← score + 1;
            else if w.count ≥ 0 then
7.              w.count ← w.count + 1;
            end if
8.          w ← NNL[i, j];
            if w.count = present then
9.              score ← score + 1;
            else if w.count ≥ 0 then
10.             w.count ← w.count + 1;
            end if
        end for
11.     w ← NNL[i, i];
        if w.count = present then
12.         score ← score + 1;
        else if w.count ≥ 0 then
13.         w.count ← w.count + 1;
        end if
14.     SCONFL[i] = score/i²;
    end for
    {Reset the counts to their default value.}
    for i = 1 to m do
15.     QNL[i].count ← ∅;
    end for
```

【図７】

アイテムq_i	SASH レベル	パッチ
i=n−1	0	$NN(R_{0,q(n-1)},m)$
i=n−2	0	$NN(R_{0,q(n-2)},m)$
i=n−3	0	$NN(R_{0,q(n-3)},m)$
⋮	⋮	⋮
i=n/2−1	1	$NN(R_{1,q(n/2-1)},m)$
⋮	⋮	⋮
i=n/4−1	2	$NN(R_{2,q(n/4-1)},m)$
⋮	⋮	⋮
i=0	h	$NN(R_{h,q(0)},m)$

【図８】

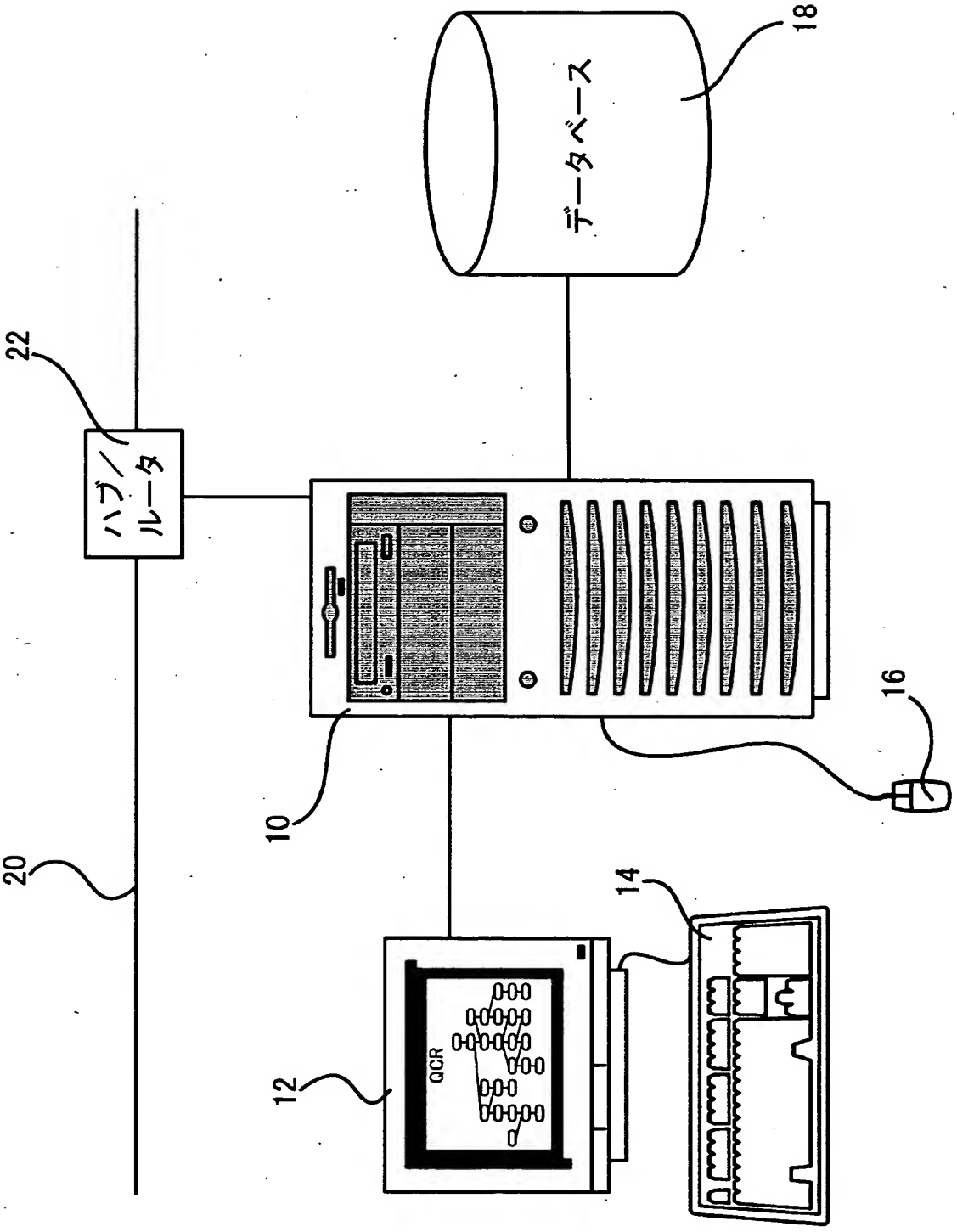
```
RefineProfile (query q;
                inner patch size k_I;
                outer patch size k_O): reordered query k_I-patch RQNL
  {Let QNL be the k_O-patch precomputed for query q.}
  {Let NNL be a list of the k_O-patches precomputed for every element of QNL.}
  {Initially, v.inpatch = false is assumed for every element v in the data set.}
  {Identify the inner patch members.}
  for i = 1 to k_I do
1.    QNL[i].inpatch ← true;
  end for
  {Initialize the confidence value CONFc of every patch element to zero.}
  for i = 1 to k_O do
2.    CONFc[i] ← 0;
  end for
  {For each element of the outer patch, count the number of elements
    of their k-nearest-neighbor sets shared with that of q.}
  for i = 1 to k_O do
    for j = 1 to k_I do
3.      w ← NNL[i, j];
        if w.inpatch = true then
4.          CONFc[i] ← CONFc[i] + 1;
        end if
    end for
5.  CONFc[i] ← CONFc[i]/k_O;
  end for
  {Reorder the outer patch elements according to their confidence values, from highest to lowest.}
6. RQNL ← sort(QNL, CONFc, k_O);
  {Reset the patch membership indicators to their default values.}
  for i = 1 to k_I do
7.    QNL[i].inpatch ← false;
  end for
```

【図９】

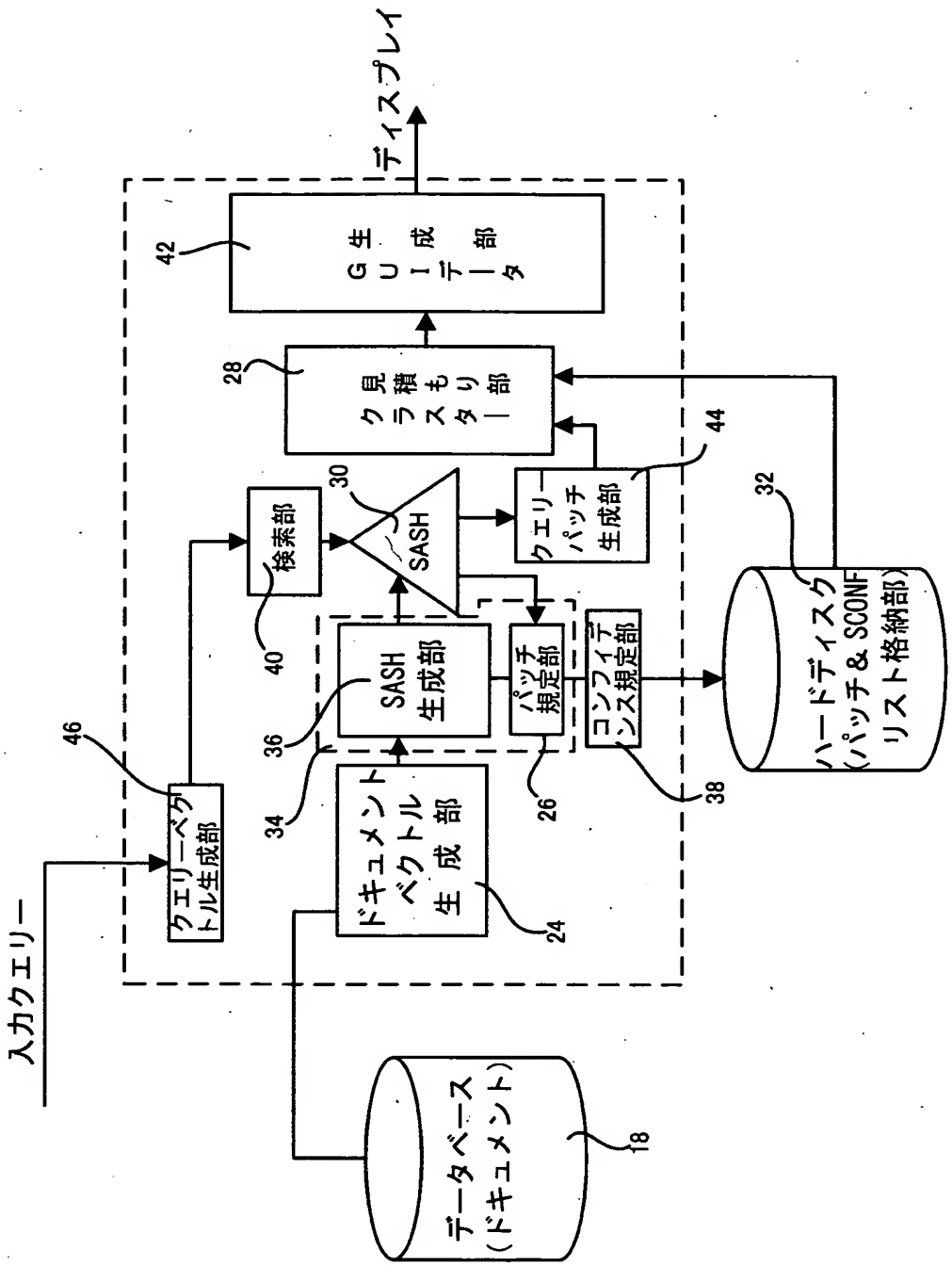
PatchCluster (data set S;
RSCM parameters a, b, $m = \varphi(b)$;
Thresholds α, β, γ, δ): query cluster graph G

1. Randomly partition the set S into subsets S_t of approximate size $\frac{|S|}{2^t}$, for $0 \leq t \leq h = \lfloor \log_2 |S| \rfloor$.
2. For all $0 \leq t \leq h$ do:
   (a) For every element $v \in S_t$, compute nearest-neighbor patches $\mathbf{NN}(R_t, v, m)$, where $R_t = \bigcup_{i \geq t} S_i$.
   (b) For each element $v_{t,i} \in S_t$, compute the optimal query cluster size $k(v_{t,i})$ maximizing $\mathbf{RSCONF}(\mathbf{NN}(R_t, v_{t,i}, k), \varphi)$, for values of k between a and b
   The ranked collection of patches
   $$\mathcal{C}_t = \langle C_{t,i} | \, i < j \Rightarrow \mathbf{RSCONF}(C_{t,i}, \varphi) \geq \mathbf{RSCONF}(C_{t,j}, \varphi) \rangle$$
   form the candidates for the query clusters associated with sample $R_t \subseteq S$, where $C_{t,i} = \mathbf{NN}(R_t, v_{t,i}, k(v_{t,i}))$ and $C_{t,j} = \mathbf{NN}(R_t, v_{t,j}, k(v_{t,j}))$.
   (c) Let Q_t be a list of patches of $\mathcal{C}_t$ that have been confirmed as query clusters of R_t. Initially, Q_t is empty.
   (d) For all $1 \leq i \leq |\mathcal{C}_t|$ do:
      i. If $\mathbf{RSCONF}(C_{t,i}, \varphi) < \alpha$, then break from the loop.
      ii. For all $w \in C_{t,i}$ do: if $\mathbf{NN}(R_t, w, k) \notin |\mathcal{C}_t|$ for any value of k, or failing that, if $\max\{\mathbf{CONF}(\mathbf{NN}(R_t, w, k), C_{t,i}), \mathbf{CONF}(C_{t,i}, \mathbf{NN}(R_t, w, k))\} < \beta$, then add $C_{t,i}$ to the list Q_t.
3. Let h' be the largest index for which $|Q_{h'}| > 0$. Let $\{C_{t,j}\}$ be the set of patches comprising Q_t, where $C_{t,j} = \mathbf{NN}(R_t, q_{t,j}, k(q_{t,j}))\}$, for all $0 \leq t \leq h'$. Initialize the node set of the query cluster graph G to contain these patches, one patch per node.
4. For all $\delta \leq t \leq h'$, all $1 \leq j \leq |Q_t|$, and all $\max\{0, t - \delta\} \leq s \leq t$, do:
   (a) Compute $C'_{t,j} = \mathbf{NN}(R_s, q_{t,j}, 2^{t-s} k(q_{t,j}))\}$.
   (b) For all $1 \leq i \leq |Q_s|$, if $C_{s,i} \neq C'_{t,j}$ and $\max\{\mathbf{CONF}(C_{s,i}, C'_{t,j}), \mathbf{CONF}(C'_{t,j}, C_{s,i})\} \geq \gamma$, then introduce the edges $(C_{s,i}, C_{t,j})$ and $(C_{t,j}, C_{s,i})$ into G, with weights $\mathbf{CONF}(C_{s,i}, C'_{t,j})$ and $\mathbf{CONF}(C'_{t,j}, C_{s,i})$, respectively.

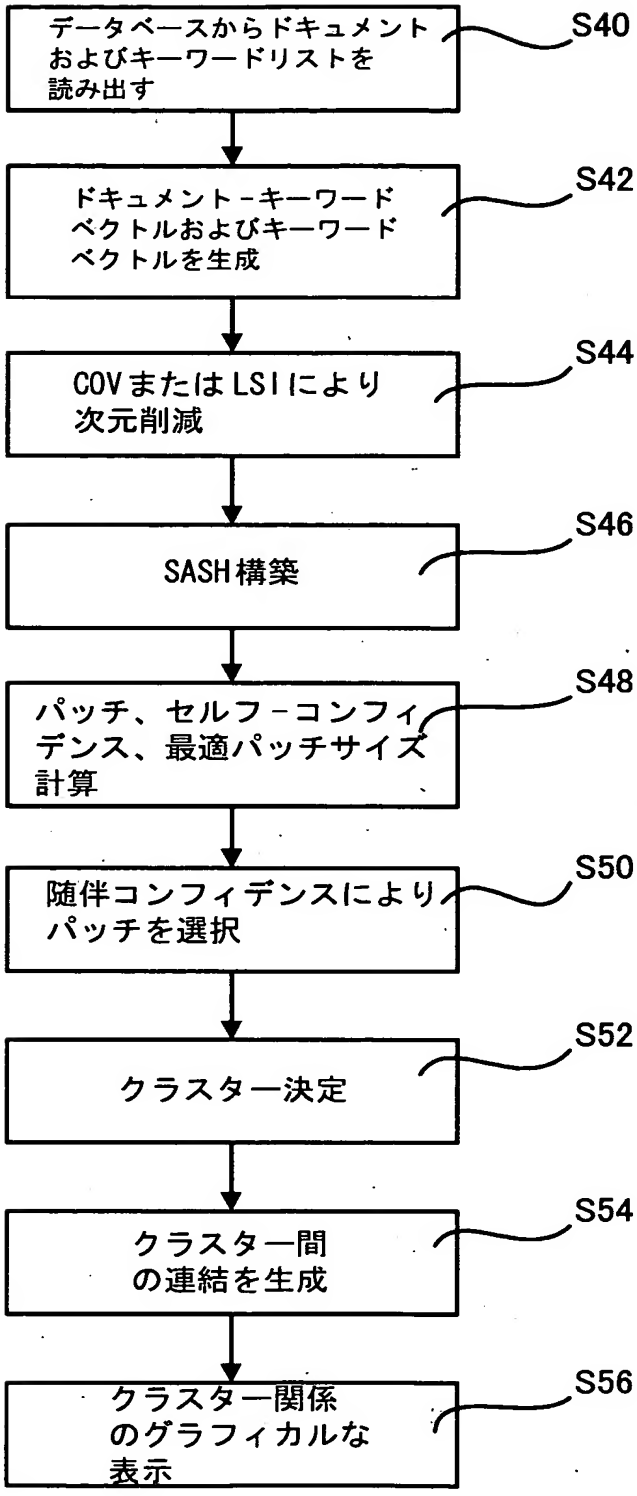
【図１０】



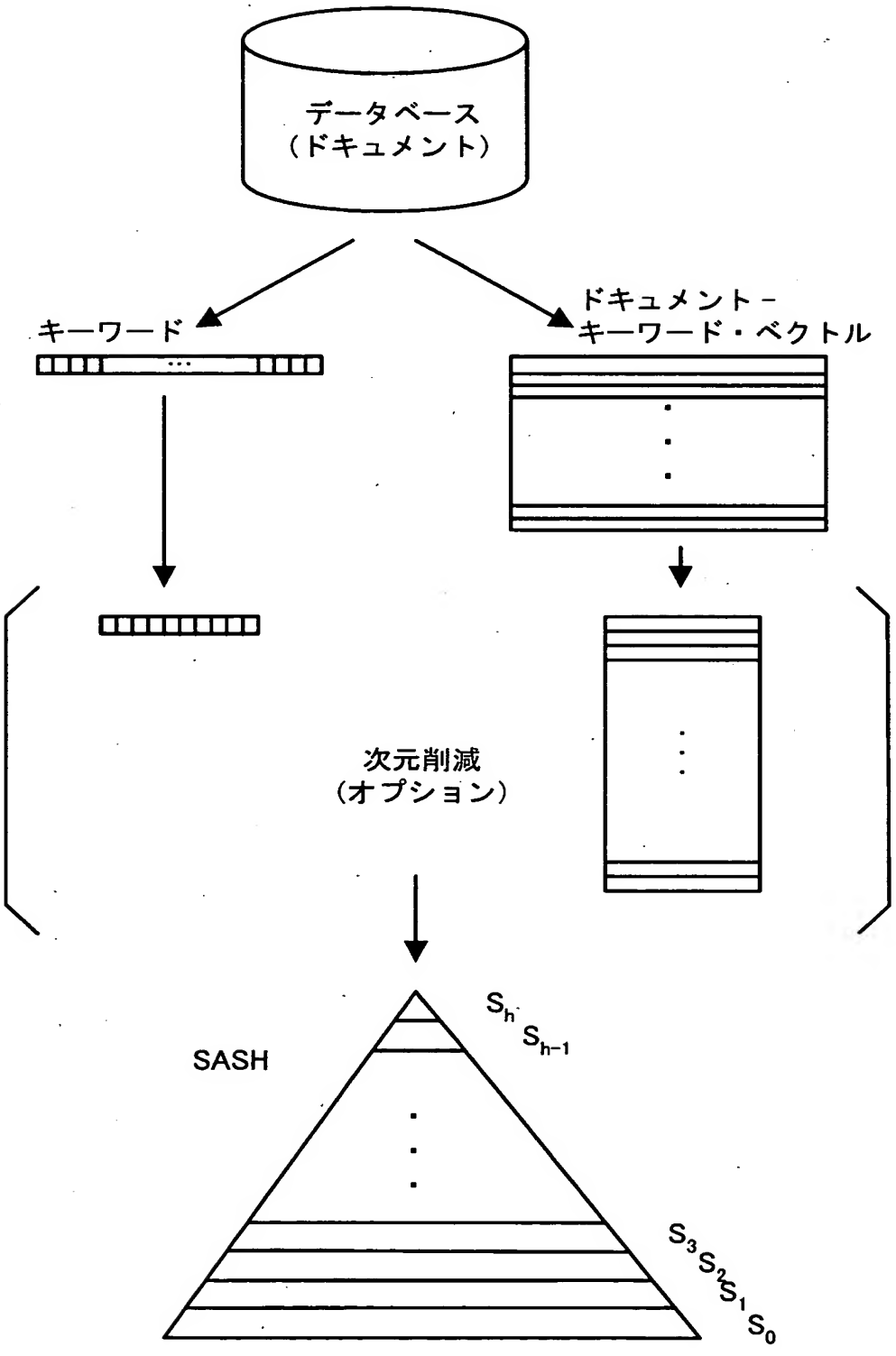
【図１１】



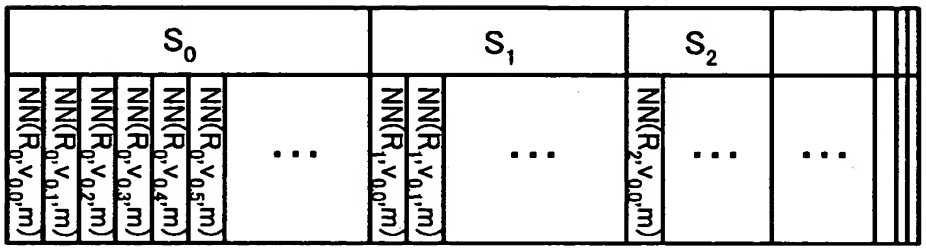
【図１２】



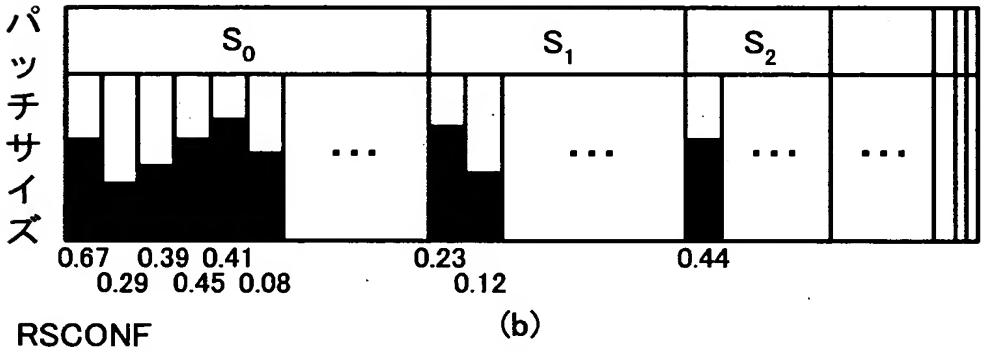
【図１３】



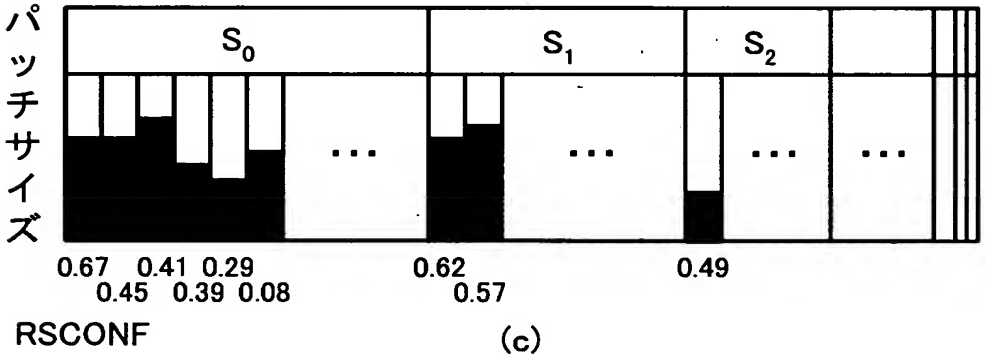
【図１４】



(a)

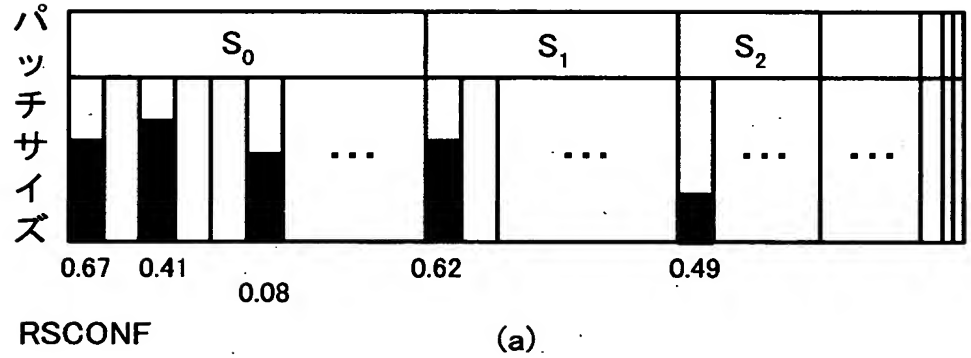


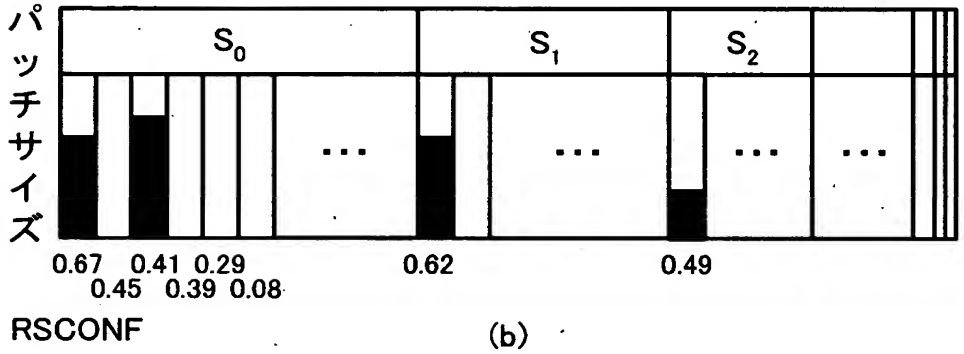
(b)



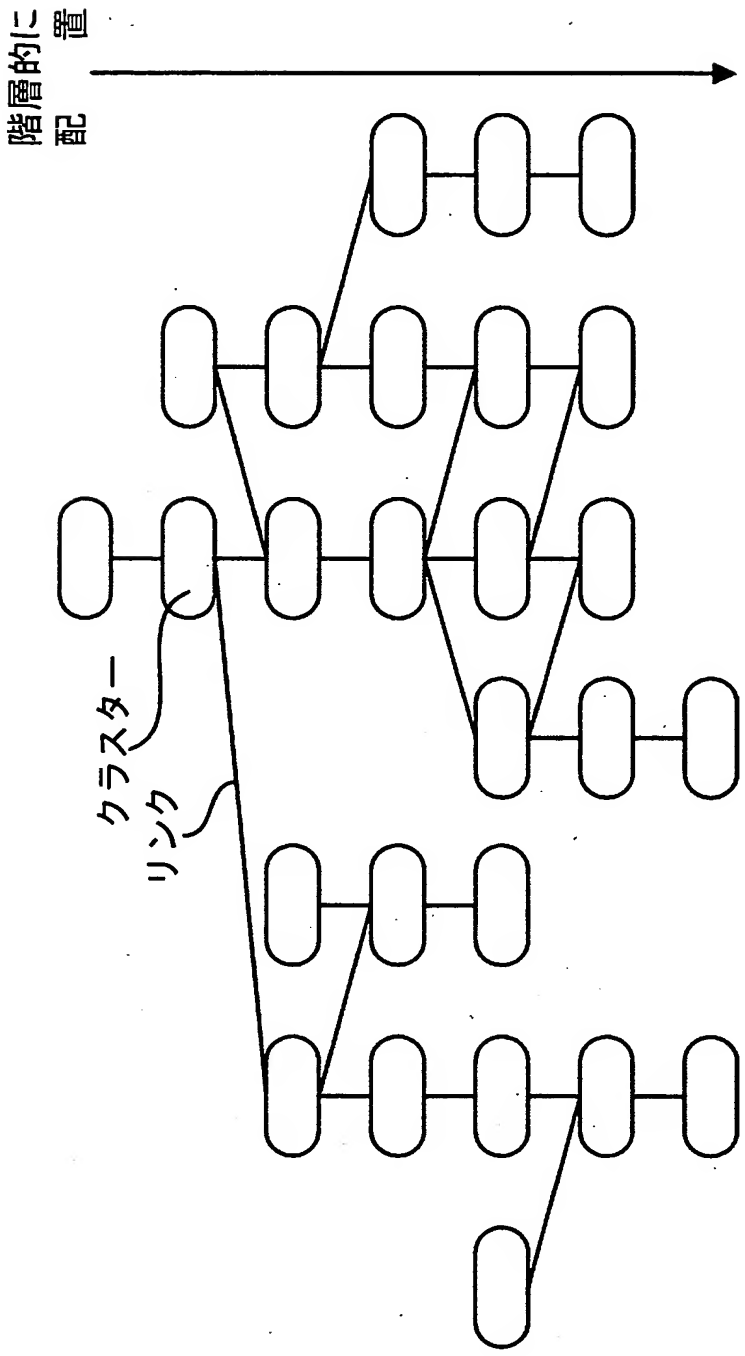
(c)

【図１５】

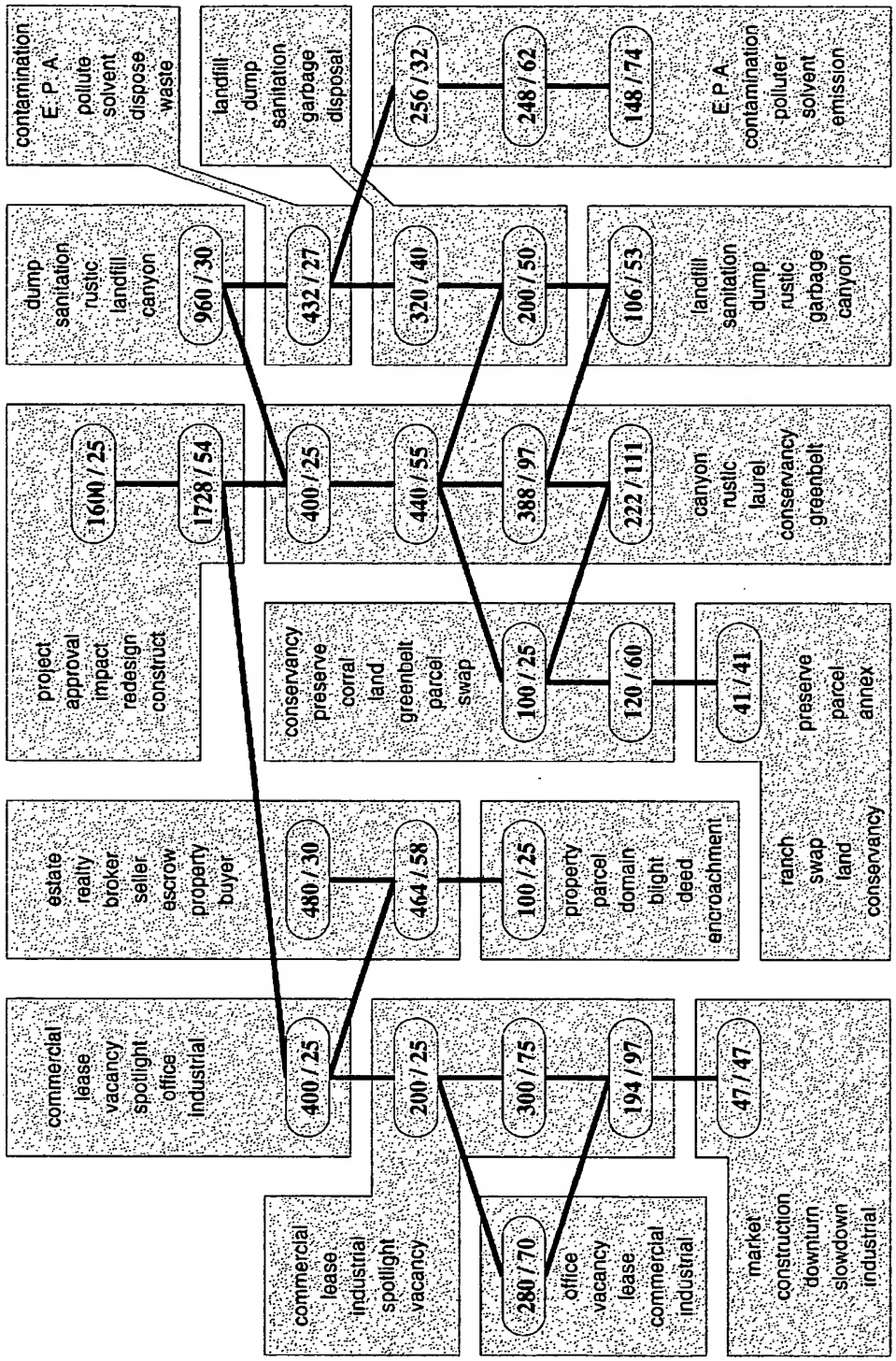




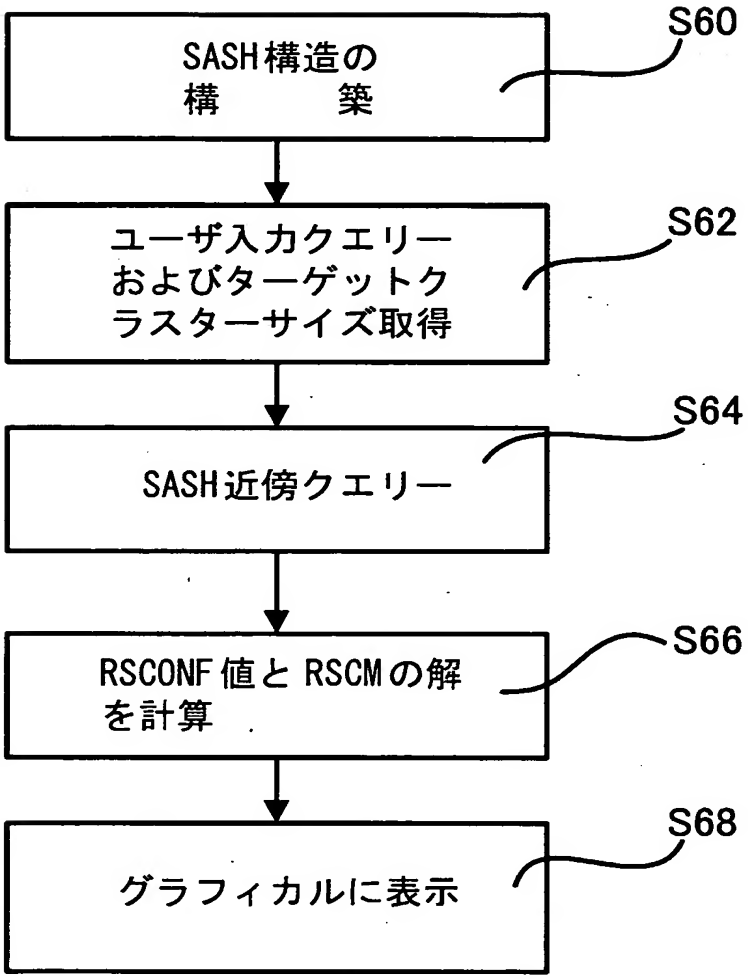
【図１６】



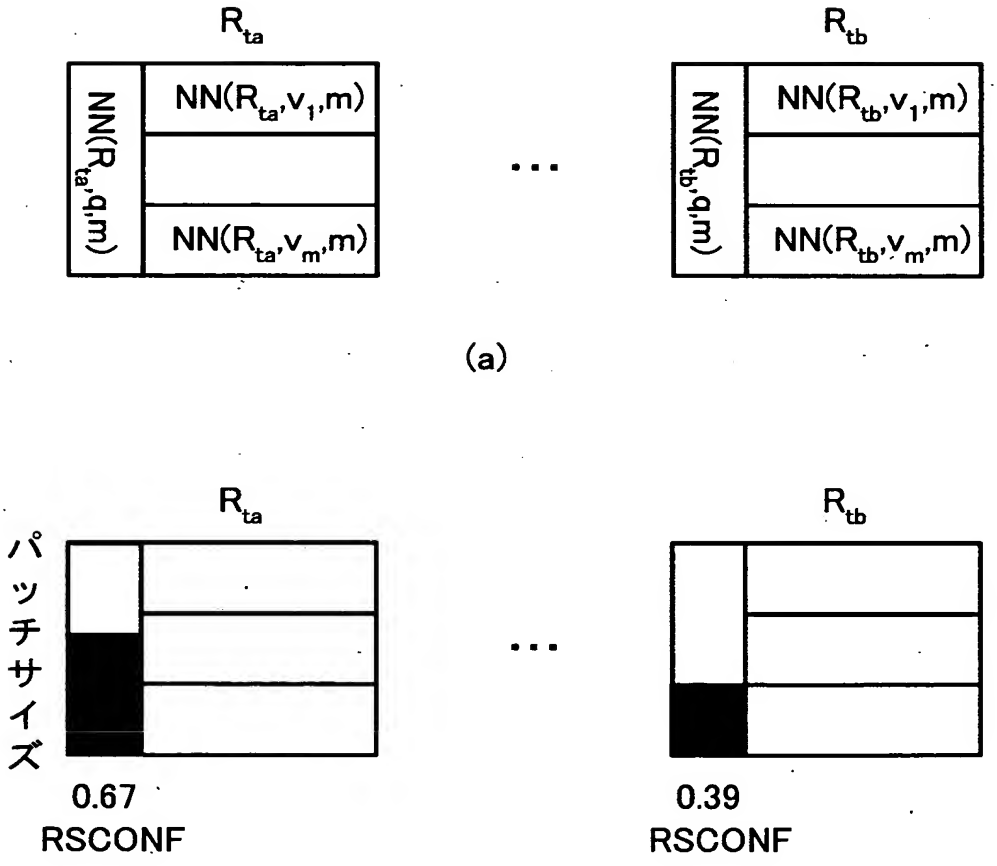
【図１７】



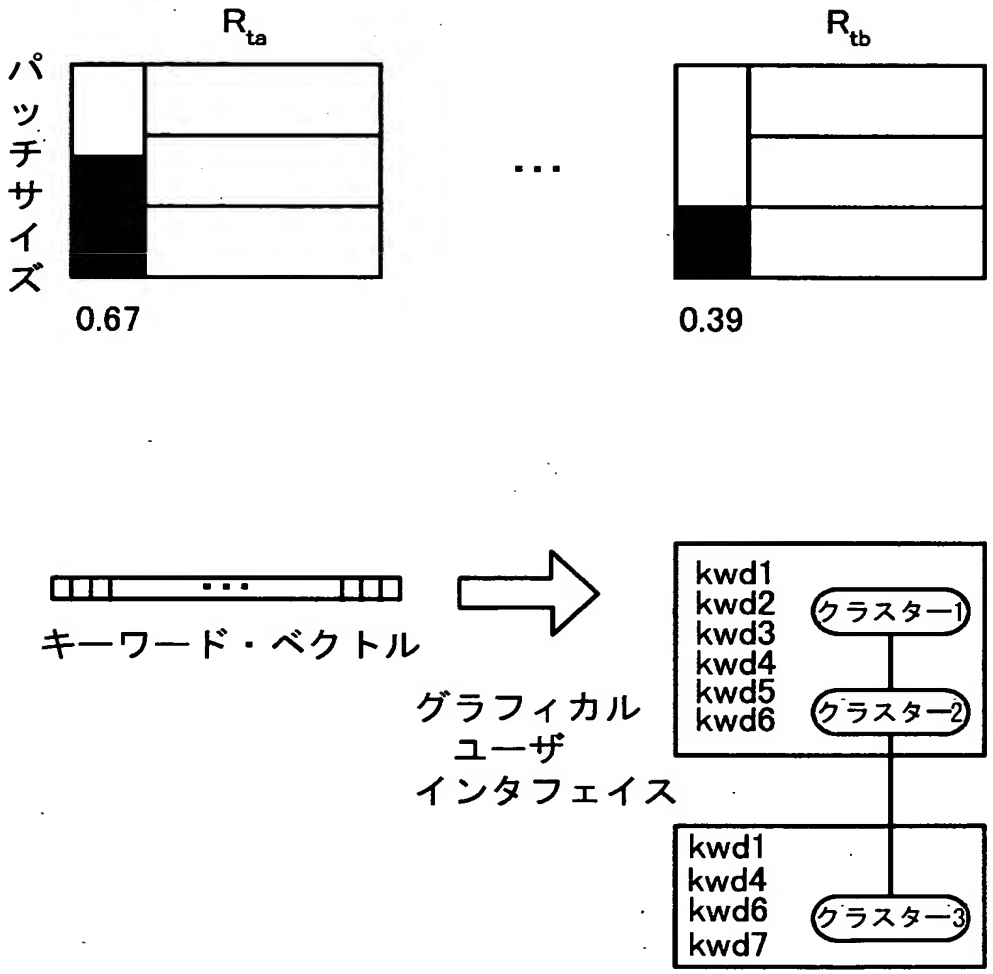
【図１８】


SASH構造の
構　　築
S60
ユーザ入力クエリー
およびターゲットク
ラスターサイズ取得
S62
SASH近傍クエリー
S64
RSCONF値とRSCMの解
を計算
S66
グラフィカルに表示
S68

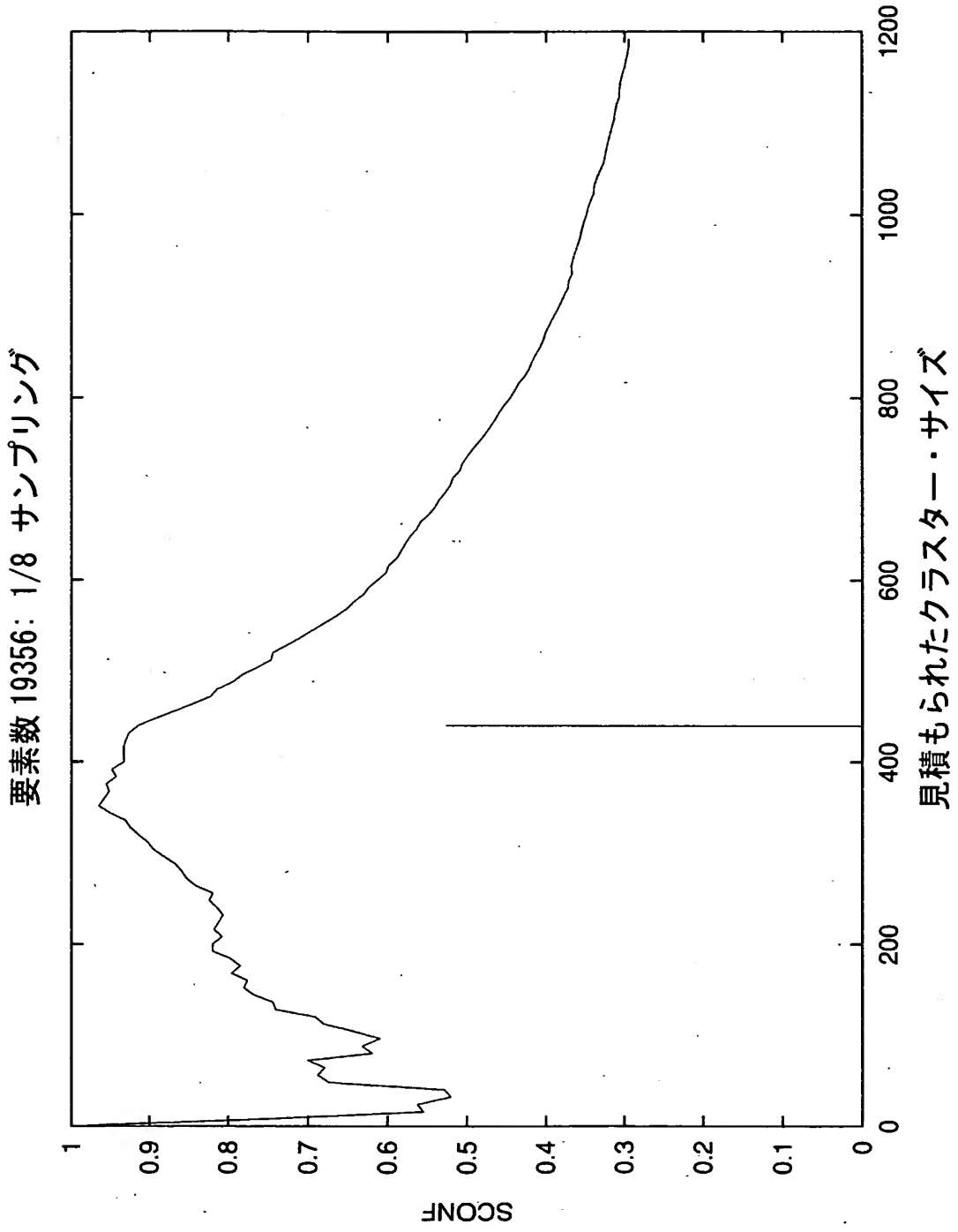
【図１９】



【図２０】



【図２１】


要素数 19356: 1/8 サンプリング
SCONF
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
200
400
600
800
1000
1200
見積もられたクラスター・サイズ

【書類名】　要約書

【要約】

【課題】　情報検索のためのデータ構造を生成するコンピュータ・システム、そのための方法、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラム、情報検索のためのデータ構造を生成するコンピュータ実行可能なプログラムを記憶したコンピュータ可読な記憶媒体、情報検索システム、およびグラフィカル・ユーザ・インタフェイス・システムを提供する。

【解決手段】　データベースに格納されたドキュメントの情報検索のためのデータ構造を生成するコンピュータ・システムは、階層構造における所定の類似性を有するノード・グループを生成するための近傍パッチ生成部３４を含む。近傍パッチ生成部３４は、ドキュメント－キーワード・ベクトルに対して階層構造を生成させる階層生成部３６と、パッチ規定部２６とを含んでいる。コンピュータ・システムはまた、パッチの間の類似性を使用してドキュメント－キーワード・ベクトルのクラスター・データを生成するためのクラスター見積もり部２８を含んで構成されている。

【選択図】　図１１

認定・付加情報

特許出願の番号	特願２００２－３６８２７６
受付番号	５０３００２６１０２３
書類名	翻訳文提出書
担当官	佐々木　吉正　　２４２４
作成日	平成１５年　２月２４日

＜認定情報・付加情報＞

【特許出願人】

【識別番号】　390009531

【住所又は居所】　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　ニュー　オーチャード　ロード

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】　100086243

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内

【氏名又は名称】　坂口　博

【代理人】

【識別番号】　100091568

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内

【氏名又は名称】　市位　嘉宏

【代理人】

【識別番号】　100108501

【住所又は居所】　神奈川県大和市下鶴間１６２３番１４　日本アイ・ビー・エム株式会社　知的所有権

【氏名又は名称】　上野　剛史

【復代理人】　申請人

【識別番号】　100110607

【住所又は居所】　神奈川県大和市中央林間３丁目４番４号　サクライビル４階　間山国際特許事務所

【氏名又は名称】　間山　進也

次頁無

出　願　人　履　歴　情　報

識別番号　　〔３９０００９５３１〕

１．変更年月日　　２００２年　６月　３日
［変更理由］　　住所変更
住　所　　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
氏　名　　インターナショナル・ビジネス・マシーンズ・コーポレーション